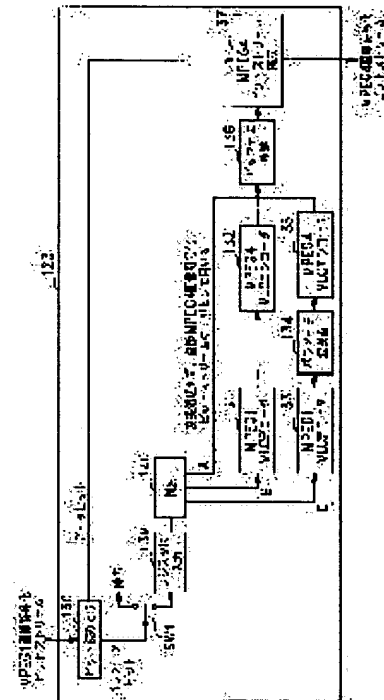


(43)Date of publication of application : 31.08.2001

(72)Inventor : ZUU IIWEN
YAGASAKI YOICHI
NAGUMO TAKEFUMI



7/14/2005

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(51) Int.Cl. ⁷	識別記号	F I	テーマコード [*] (参考)
H 0 4 N	7/30	H 0 3 M 7/38	5 C 0 5 9
H 0 3 M	7/38	H 0 4 N 7/133	Z 5 J 0 6 4
H 0 4 N	7/32	7/137	Z

審査請求 未請求 請求項の数22 O L (全 28 頁)

(21) 出願番号 特願2000-50523 (P2000-50523)

(22) 出願日 平成12年2月22日 (2000.2.22)

(71) 出願人 000002185

ソニー株式会社

東京都品川区北品川6丁目7番35号

(72) 発明者 ズー イーウェン

東京都品川区北品川6丁目7番35号 ソニー株式会社内

(72) 発明者 矢ヶ崎 陽一

東京都品川区北品川6丁目7番35号 ソニー株式会社内

(74) 代理人 100067736

弁理士 小池 晃 (外2名)

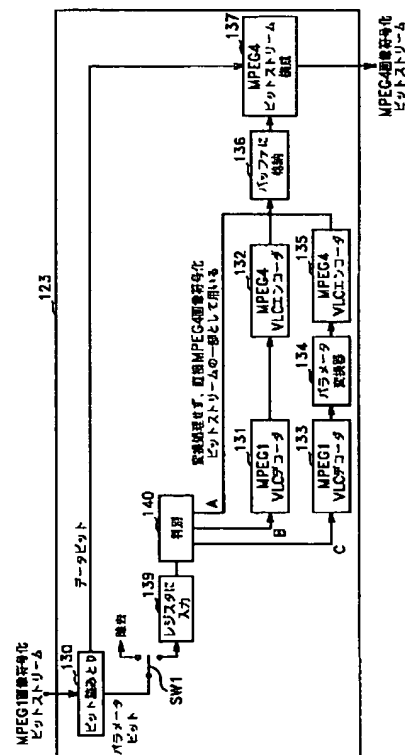
最終頁に続く

(54) 【発明の名称】 符号化フォーマット変換装置及び方法、符号化信号生成装置及び方法

(57) 【要約】

【課題】 例えばMPEG1画像符号化ビットストリームをMPEG4画像符号化ビットストリームに変換する場合に、信号品質を劣化させずに処理量を低減し、且つ、時間遅延を少なくし、さらに装置構成の大規模化をも防止可能とする。

【解決手段】 ビット読み取り器130は、MPEG1画像符号化ビットストリームからMPEG1パラメータを読み取る。切換処理部SW1とレジスタ群139及び判別処理部140は、そのMPEG1パラメータを当該パラメータの値に応じて、MPEG1パラメータをそのままMPEG4パラメータとして用いるAと、MPEG1パラメータをMPEG1VLCデコードした後にMPEG4VLC円コードするBと、MPEG1パラメータをMPEG1VLCデコードした後にMPEG4パラメータに変換し更にMPEG4VLCエンコードするCとに分類する。



【特許請求の範囲】

【請求項 1】 第 1 の符号化方式により符号化された第 1 の符号化信号を、上記第 1 の符号化方式とは異なる第 2 の符号化方式による第 2 の符号化信号に変換する符号化フォーマット変換装置において、

上記第 1 の符号化信号に含まれる第 1 の符号化方式のパラメータを、そのパラメータの値に応じて上記第 2 の符号化方式のパラメータに変換するパラメータ変換手段と、

上記パラメータ変換手段により得られた第 2 の符号化方式のパラメータを用いて、上記第 1 の符号化信号から第 2 の符号化信号を生成する信号生成手段とを有することを特徴とする符号化フォーマット変換装置。

【請求項 2】 上記パラメータ変換手段は、上記第 1 の符号化信号から上記第 1 の符号化方式のパラメータを抽出する抽出手段と、当該第 1 の符号化方式のパラメータを当該パラメータの値に応じて分類する分類手段と、上記第 1 の符号化方式のパラメータに対して上記分類の結果に基づく変換処理を施す変換処理手段とを有することを特徴とする請求項 1 記載の符号化フォーマット変換装置。

【請求項 3】 上記変換処理手段は、上記分類の結果に応じて、上記第 1 の符号化方式のパラメータをそのまま第 2 の符号化方式のパラメータとして用いる第 1 の変換処理と、上記第 1 の符号化方式のパラメータを当該第 1 の符号化方式に対応する復号化方式により復号化した後に第 2 の符号化方式にて符号化する第 2 の変換処理と、上記第 1 の符号化方式のパラメータを当該第 1 の符号化方式に対応する復号化方式により復号化した後に第 2 の符号化方式のパラメータに変換し当該第 2 の符号化方式のパラメータを第 2 の符号化方式にて符号化する第 3 の変換処理とを、選択的に使用することを特徴とする請求項 2 記載の符号化フォーマット変換装置。

【請求項 4】 上記信号生成手段は、当該第 2 の符号化信号を復号化する際に使用する同期信号を、上記第 2 の符号化信号を構成する符号化信号列の所定の処理単位の先頭に配置する機能を有することを特徴とする請求項 1 記載の符号化フォーマット変換装置。

【請求項 5】 上記第 2 の符号化方式が M P E G 2 画像符号化方式であるとき、

上記信号生成手段は、画像の V O P ヘッダの先頭以外の行の先頭に上記同期信号としてリシンクマーカを配置することを特徴とする請求項 4 記載の符号化フォーマット変換装置。

【請求項 6】 上記第 1 の符号化方式に対応する復号化方式の逆量子化方法と、上記第 2 の符号化方式に対応する復号化方式の逆量子化方法とが異なるとき、
上記パラメータ変換手段は、上記第 1 の符号化信号を逆量子化した信号の主要部の値と、上記第 2 の符号化信号を逆量子化した信号の主要部の値とを一致させるよう

に、上記第 1 の符号化方式の量子化パラメータを上記第 2 の符号化方式の量子化パラメータに変換する機能を有することを特徴とする請求項 1 記載の符号化フォーマット変換装置。

【請求項 7】 上記第 1 の符号化信号と第 2 の符号化信号とが、それぞれ所定符号化単位内で完結した符号化が行われた第 1 の信号と所定符号化単位間の予測による符号化が行われた第 2 の信号とを少なくとも含むとき、
上記信号生成手段は、上記第 1 の符号化信号内の上記第 1 の信号を、上記第 2 の符号化方式の第 2 の信号に直接変換する機能を有することを特徴とする請求項 1 記載の符号化フォーマット変換装置。

【請求項 8】 第 1 の符号化方式により符号化された第 1 の符号化信号を当該第 1 の符号化方式とは異なる第 2 の符号化方式による第 2 の符号化信号に変換する符号化フォーマット変換装置に対して、上記第 1 の符号化信号を生成して供給する符号化信号生成装置であって、
上記第 1 の符号化方式にて使用される第 1 の量子化方法と上記第 2 の符号化方式にて使用される第 2 の量子化方法とが異なるとき、上記第 1 の量子化方法で用いるパラメータに代え、上記第 2 の量子化方法で用いるパラメータを上記第 1 の符号化方式により符号化された信号に含めて、上記符号化フォーマット変換装置に供給する第 1 の符号化信号を生成するパラメータ変更手段を有することを特徴とする符号化信号生成装置。

【請求項 9】 動き補償予測を用いた第 1 の画像符号化方式により符号化された第 1 の符号化信号を、動き補償予測を用いると共に上記第 1 の画像符号化方式とは異なる第 2 の画像符号化方式による第 2 の符号化信号に変換する符号化フォーマット変換装置に対して、上記第 1 の符号化信号を生成して供給する符号化信号生成装置であって、

上記第 1 の符号化信号を生成する際に用いた色差の動きベクトルと、上記第 2 の符号化信号を生成する際に用いた色差の動きベクトルとが異なるとき、上記第 1 の符号化信号での色差の動きベクトルと上記第 2 の符号化信号での色差の動きベクトルとが等しくなるように、上記第 1 の符号化信号の輝度の動きベクトルを上記第 2 の符号化信号の輝度の動きベクトルに変換する動きベクトル変換手段を有することを特徴とする符号化信号生成装置。

【請求項 1 0】 上記第 1 の画像符号化方式は M P E G 1 若しくは M P E G 2 画像符号化方式であり、上記第 2 の画像符号化方式は M P E G 4 画像符号化方式であり、
上記動きベクトル変換手段は、上記第 1 の符号化信号を生成する際に用いた輝度の動きベクトルが 4 で割って 1 余る値であるとき、当該輝度の動きベクトルを 4 で割って余りが 1 以外の値を持つ動きベクトルに変換する機能を有することを特徴とする請求項 9 記載の符号化信号生成装置。

【請求項 1 1】 上記第 1 の画像符号化方式は M P E G

1 若しくはMPEG 2 画像符号化方式であり、上記第2の画像符号化方式はMPEG 4 画像符号化方式であり、上記動きベクトル変換手段は、上記第1の符号化信号を生成する際に用いた輝度の動きベクトルが1であるとき、当該動きベクトルを0に変換する機能を有することを特徴とする請求項9記載の符号化信号生成装置。

【請求項12】 第1の符号化方式により符号化された第1の符号化信号を、上記第1の符号化方式とは異なる第2の符号化方式による第2の符号化信号に変換する符号化フォーマット変換方法において、
上記第1の符号化信号に含まれる第1の符号化方式のパラメータを、そのパラメータの値に応じて上記第2の符号化方式のパラメータに変換し、
上記パラメータ変換により得られた第2の符号化方式のパラメータを用いて、上記第1の符号化信号から第2の符号化信号を生成することを特徴とする符号化フォーマット変換方法。

【請求項13】 上記パラメータ変換の際には、上記第1の符号化信号から上記第1の符号化方式のパラメータを抽出し、当該第1の符号化方式のパラメータを当該パラメータの値に応じて分類し、上記第1の符号化方式のパラメータに対して上記分類の結果に基づく変換処理を施すことを特徴とする請求項12記載の符号化フォーマット変換方法。

【請求項14】 上記分類の結果に基づく変換処理の際には、上記分類の結果に応じて、上記第1の符号化方式のパラメータをそのまま第2の符号化方式のパラメータとして用いる第1の変換処理と、上記第1の符号化方式のパラメータを当該第1の符号化方式に対応する復号化方式により復号化した後に第2の符号化方式にて符号化する第2の変換処理と、上記第1の符号化方式のパラメータを当該第1の符号化方式に対応する復号化方式により復号化した後に第2の符号化方式のパラメータに変換し当該第2の符号化方式のパラメータを第2の符号化方式にて符号化する第3の変換処理とを、選択的に使用することを特徴とする請求項13記載の符号化フォーマット変換方法。

【請求項15】 上記第2の符号化信号の生成の際には、上記第2の符号化信号を復号化する際に使用する同期信号を、上記第2の符号化信号を構成する符号化信号列の所定の処理単位の先頭に配置することを特徴とする請求項12記載の符号化フォーマット変換方法。

【請求項16】 上記第2の符号化方式がMPEG 2 画像符号化方式であるとき、
上記第2の符号化信号の生成の際には、画像のVOPヘッダの先頭以外の行の先頭に上記同期信号としてリシンクマークを配置することを特徴とする請求項15記載の符号化フォーマット変換方法。

【請求項17】 上記第1の符号化方式に対応する復号化方式の逆量子化方法と、上記第2の符号化方式に対応

する復号化方式の逆量子化方法とが異なるとき、
上記パラメータの変換の際には、上記第1の符号化信号を逆量子化した信号の主要部の値と、上記第2の符号化信号を逆量子化した信号の主要部の値とを一致させるように、上記第1の符号化方式の量子化パラメータを上記第2の符号化方式の量子化パラメータに変換することを特徴とする請求項12記載の符号化フォーマット変換方法。

【請求項18】 上記第1の符号化信号と第2の符号化信号とが、それぞれ所定符号化単位内で完結した符号化が行われた第1の信号と所定符号化単位間の予測による符号化が行われた第2の信号とを少なくとも含むとき、
上記第2の符号化信号の生成の際には、上記第1の符号化信号内の上記第1の信号を、上記第2の符号化方式の第2の信号に直接変換することを特徴とする請求項12記載の符号化フォーマット変換方法。

【請求項19】 第1の符号化方式により符号化された第1の符号化信号を当該第1の符号化方式とは異なる第2の符号化方式による第2の符号化信号に変換する符号化フォーマット変換装置に対して、上記第1の符号化信号を生成して供給する符号化信号生成方法であって、
上記第1の符号化方式にて使用される第1の量子化方法と上記第2の符号化方式にて使用される第2の量子化方法とが異なるとき、上記第1の量子化方法で用いるパラメータに代え、上記第2の量子化方法で用いるパラメータを上記第1の符号化方式により符号化された信号に含めて、上記符号化フォーマット変換装置に供給する第1の符号化信号を生成することを特徴とする符号化信号生成方法。

【請求項20】 動き補償予測を用いた第1の画像符号化方式により符号化された第1の符号化信号を、動き補償予測を用いると共に上記第1の画像符号化方式とは異なる第2の画像符号化方式による第2の符号化信号に変換する符号化フォーマット変換装置に対して、上記第1の符号化信号を生成して供給する符号化信号生成方法であって、

上記第1の符号化信号を生成する際に用いた色差の動きベクトルと、上記第2の符号化信号を生成する際に用いた色差の動きベクトルとが異なるとき、上記第1の符号化信号での色差の動きベクトルと上記第2の符号化信号での色差の動きベクトルとが等しくなるように、上記第1の符号化信号の輝度の動きベクトルを上記第2の符号化信号の輝度の動きベクトルに変換することを特徴とする符号化信号生成方法。

【請求項21】 上記第1の画像符号化方式はMPEG 1 若しくはMPEG 2 画像符号化方式であり、上記第2の画像符号化方式はMPEG 4 画像符号化方式であり、
上記動きベクトルの変換の際には、上記第1の符号化信号を生成する際に用いた輝度の動きベクトルが4で割って1余る値であるとき、当該輝度の動きベクトルを4で

割って余りが1以外の値を持つ動きベクトルに変換することを特徴とする請求項20記載の符号化信号生成方法。

【請求項22】 上記第1の画像符号化方式はMPEG1若しくはMPEG2画像符号化方式であり、上記第2の画像符号化方式はMPEG4画像符号化方式であり、上記動きベクトルの変換の際には、上記第1の符号化信号を生成する際に用いた輝度の動きベクトルが1であるとき、当該動きベクトルを0に変換することを特徴とする請求項20記載の符号化信号生成方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、符号化フォーマット変換装置及び方法、符号化信号生成装置及び方法に関し、特に、例えば光磁気ディスクや磁気テープ、フラッシュメモリ等の記録媒体に記録し、もしくは衛星放送で伝送し、それを光磁気ディスクや磁気テープ、フラッシュメモリ等の記録媒体に再記録、あるいはテレビ会議システムやテレビ電話システム、インターネット、携帯電話等低ビットレート伝送路を介して送信側から受信側に伝送し、受信側において、必要に応じたフォーマット変換等を行い、これを表示、伝送する場合などに用いて好適な符号化フォーマット変換装置及び方法、符号化信号生成装置及び方法に関する。

【0002】

【従来の技術】入力された画像符号化ビットストリームを、異なる画像符号化フォーマットの画像符号化ビットストリームに変換する、従来の画像符号化フォーマット変換装置の概略構成を図39に示す。なお、この図39に例示した画像符号化フォーマット変換装置315は、入力されたMPEG1 (Moving Picture image coding Experts Group) 画像符号化ビットストリームを、MPEG4画像符号化ビットストリームへ変換する装置である。また、図39の例では、MPEG1画像符号化ビットストリームを生成するMPEG1画像符号化装置310と、MPEG1画像符号化ビットストリームを記録する記録器311、MPEG1画像符号化ビットストリームを伝送する伝送器312も示す。

【0003】この図39に示すシステムにおいて、MPEG1画像符号化装置310は、入力された画像信号をMPEG1画像符号化方式（以下、適宜MPEG1とする）に従って符号化し、MPEG1画像符号化ビットストリームを生成する。当該MPEG1画像符号化ビットストリームは、直接に、或いは記録器311に一旦記録された後に再生されて、若しくは伝送器312を介して伝送されて、画像符号化フォーマット変換装置315に inputs する。

【0004】画像符号化フォーマット変換装置315は、MPEG1画像復号化器313と、MPEG4画像符号化器314とからなる。

【0005】MPEG1画像復号化器313は、入力されたMPEG1画像符号化ビットストリームを、MPEG1画像復号化方式に従って復号して画像信号を復元する。当該復元された画像信号はMPEG4画像符号化器314に送られる。

【0006】MPEG4画像符号化器314は、上記復元された画像信号をMPEG4画像符号化方式（以下、適宜MPEG4とする）に従って符号化し、MPEG4画像符号化ビットストリームを生成する。このMPEG4画像符号化ビットストリームは、当該図39の画像符号化フォーマット変換装置315から出力される。

【0007】

【発明が解決しようとする課題】上述の従来の画像符号化フォーマット変換装置315は、図39に示したように、MPEG1画像復号化器313でMPEG1画像復号化方式により復元された画像信号を、MPEG4画像符号化器314でMPEG4画像符号化方式により符号化し、MPEG4画像符号化ビットストリームを出力するようになされている。

【0008】すなわち、この従来の画像符号化フォーマット変換装置315のプロセスには、MPEG1画像復号化器313で画像信号を復元するプロセスと、MPEG4画像符号化器314で画像信号を符号化するプロセスとが含まれている。

【0009】特に、MPEG4画像符号化器314では、入力された画像信号を符号化する際に、動きベクトルを検出するなど、処理量が膨大となるプロセスが含まれている。このため、従来の画像符号化フォーマット変換装置315では、画像のリアルタイムでの処理が困難となり、時間遅延が発生するという問題がある。また、装置が大規模になるという問題もある。

【0010】そこで、本発明は、このような実情に鑑みてなされたものであり、例えば、MPEG1画像符号化ビットストリームのような第1の符号化信号をMPEG4画像符号化ビットストリームのような第2の符号化信号に変換する場合において、信号品質を劣化させずに処理量を低減し、且つ、時間遅延を少なくし、さらに装置構成の大規模化をも防止可能とする符号化フォーマット変換装置及び方法と、第1の符号化信号を生成する符号化信号生成装置及び方法を提供することを目的とする。

【0011】

【課題を解決するための手段】本発明の符号化フォーマット変換装置は、第1の符号化方式により符号化された第1の符号化信号を、上記第1の符号化方式とは異なる第2の符号化方式による第2の符号化信号に変換する符号化フォーマット変換装置において、上記第1の符号化信号に含まれる第1の符号化方式のパラメータを、そのパラメータの値に応じて上記第2の符号化方式のパラメータに変換するパラメータ変換手段と、上記パラメータ変換手段により得られた第2の符号化方式のパラメータ

を用いて、上記第1の符号化信号から第2の符号化信号を生成する信号生成手段とを有することにより、上述した課題を解決する。

【0012】本発明の符号化信号生成装置は、第1の符号化方式により符号化された第1の符号化信号を当該第1の符号化方式とは異なる第2の符号化方式による第2の符号化信号に変換する符号化フォーマット変換装置に対して、上記第1の符号化信号を生成して供給する符号化信号生成装置であって、上記第1の符号化方式にて使用される第1の量子化方法と上記第2の符号化方式にて使用される第2の量子化方法とが異なるとき、上記第1の量子化方法で用いるパラメータに代え、上記第2の量子化方法で用いるパラメータを上記第1の符号化方式により符号化された信号に含めて、上記符号化フォーマット変換装置に供給する第1の符号化信号を生成するパラメータ変更手段を有することにより、上述した課題を解決する。

【0013】本発明の符号化信号生成装置は、動き補償予測を用いた第1の画像符号化方式により符号化された第1の符号化信号を、動き補償予測を用いると共に上記第1の画像符号化方式とは異なる第2の画像符号化方式による第2の符号化信号に変換する符号化フォーマット変換装置に対して、上記第1の符号化信号を生成して供給する符号化信号生成装置であって、上記第1の符号化信号を生成する際に用いた色差の動きベクトルと、上記第2の符号化信号を生成する際に用いた色差の動きベクトルとが異なるとき、上記第1の符号化信号での色差の動きベクトルと上記第2の符号化信号での色差の動きベクトルとが等しくなるように、上記第1の符号化信号の輝度の動きベクトルを上記第2の符号化信号の輝度の動きベクトルに変換する動きベクトル変換手段を有することにより、上述した課題を解決する。

【0014】本発明の符号化フォーマット変換方法は、第1の符号化方式により符号化された第1の符号化信号を、上記第1の符号化方式とは異なる第2の符号化方式による第2の符号化信号に変換する符号化フォーマット変換方法において、上記第1の符号化信号に含まれる第1の符号化方式のパラメータを、そのパラメータの値に応じて上記第2の符号化方式のパラメータに変換し、上記パラメータ変換により得られた第2の符号化方式のパラメータを用いて、上記第1の符号化信号から第2の符号化信号を生成することにより、上述した課題を解決する。

【0015】本発明の符号化信号生成方法は、第1の符号化方式により符号化された第1の符号化信号を当該第1の符号化方式とは異なる第2の符号化方式による第2の符号化信号に変換する符号化フォーマット変換装置に対して、上記第1の符号化信号を生成して供給する符号化信号生成方法であって、上記第1の符号化方式にて使用される第1の量子化方法と上記第2の符号化方式にて

使用される第2の量子化方法とが異なるとき、上記第1の量子化方法で用いるパラメータに代え、上記第2の量子化方法で用いるパラメータを上記第1の符号化方式により符号化された信号に含めて、上記符号化フォーマット変換装置に供給する第1の符号化信号を生成することにより、上述した課題を解決する。

【0016】本発明の符号化信号生成方法は、動き補償予測を用いた第1の画像符号化方式により符号化された第1の符号化信号を、動き補償予測を用いると共に上記第1の画像符号化方式とは異なる第2の画像符号化方式による第2の符号化信号に変換する符号化フォーマット変換装置に対して、上記第1の符号化信号を生成して供給する符号化信号生成方法であって、上記第1の符号化信号を生成する際に用いた色差の動きベクトルと、上記第2の符号化信号を生成する際に用いた色差の動きベクトルとが異なるとき、上記第1の符号化信号での色差の動きベクトルと上記第2の符号化信号での色差の動きベクトルとが等しくなるように、上記第1の符号化信号の輝度の動きベクトルを上記第2の符号化信号の輝度の動きベクトルに変換することにより、上述した課題を解決する。

【0017】具体例を挙げて説明すると、本発明では、例えば、MPEG1画像符号化フォーマットのビットストリームをMPEG4画像符号化フォーマットのビットストリームに変換する際に、MPEG1画像復号化装置による画像の復号およびMPEG4画像符号化装置による画像の符号化を行わず、MPEG1画像符号化ビットストリーム上からパラメータの抽出のみを行い、これをMPEG4画像符号化方式のシンプルプロファイル (simple profile) で用いるパラメータに変換する。その後、MPEG4に変換されたパラメータをMPEG4シンプルプロファイルのシンタックス (syntax) に従い、ビットストリームに含ませてMPEG4画像符号化ビットストリームを構成して出力することにより、MPEG1からMPEG4への画像符号化フォーマット変換を行うようにしている。

【0018】

【発明の実施の形態】以下、本発明の好ましい実施の形態について、図面を参照しながら説明する。

【0019】入力された画像符号化ビットストリームを、異なる画像符号化フォーマットの画像符号化ビットストリームに変換する、本発明にかかる第1の実施の形態の画像符号化フォーマット変換装置の概略構成を図1に示す。なお、この図1に例示した画像符号化フォーマット変換装置123は、入力されたMPEG1画像符号化ビットストリームを、MPEG4画像符号化ビットストリームへ変換する装置である。また、図1の例では、MPEG1画像符号化ビットストリームを生成するMPEG1画像符号化装置120と、MPEG1画像符号化ビットストリームを記録する記録器121、MPEG1

10

20

30

40

50

画像符号化ビットストリームを伝送する伝送器122も示している。

【0020】この図1に示すシステムにおいて、MPEG1画像符号化装置120は、入力された画像信号をMPEG1画像符号化方式に従って符号化し、MPEG1画像符号化ビットストリームを生成する。当該MPEG1画像符号化ビットストリームは、直接に、或いは記録器121に一旦記録された後に再生されて、若しくは伝送器122を介して伝送されて、画像符号化フォーマット変換装置123に入力する。

【0021】画像符号化フォーマット変換装置123は、MPEG1画像符号化ビットストリームをMPEG4画像符号化ビットストリームに変換する。このMPEG4画像符号化ビットストリームは、当該図1の画像符号化フォーマット変換装置123から出力される。

【0022】以上は図1の構成の概略的な処理の流れであるが、本実施の形態の画像符号化フォーマット変換装置123では、後述するように、MPEG1画像符号化ビットストリームから、符号化の際に使用されたパラメータを抽出し、その抽出したパラメータを、MPEG4画像符号化ビットストリームを構成するのに必要なパラメータに変換し、MPEG4画像符号化ビットストリームを構成して出力する。

【0023】すなわち、画像符号化フォーマット変換装置123では、前記図39に示した従来例の構成のように、MPEG1画像符号化ビットストリームを復号化して画像信号を復元するようなことやその他解像度変換等の処理を行わず、MPEG1画像符号化ビットストリーム上からパラメータの抽出のみを行い、当該抽出したパラメータを、MPEG4のシンプルフファイル (simple profile) で用いるパラメータに変換する。その後、画像符号化フォーマット変換装置123は、当該シンプルフファイルのパラメータを用い、MPEG4のシンプルフファイルのシンタックス (syntax) に従い、MPEG4画像符号化ビットストリームを構成し、出力する。

【0024】ここで、本発明実施の形態の画像符号化フォーマット変換装置123の詳細な構成及びその動作原理を、図2を用いて説明する。

【0025】図2において、入力されたMPEG1画像符号化ビットストリームは、ビットの読み取り器130に送られる。当該ビット読み取り器130は、MPEG1画像符号化ビットストリームに含まれているパラメータ (以下、MPEG1パラメータと呼ぶ。) のビットを読み取り、当該読み取られたMPEG1パラメータを切換処理部SW1に送り、また、データビットをMPEG4ビットストリーム構成器137に送る。切換処理部SW1は、上記パラメータのビットを、予め定められた2つの動作に従って、除去、若しくはレジスタ群139に入力する。

【0026】レジスタ群139に入力されたパラメータは、判別処理部140により、A、B、Cの3種の何れかに判別 (分類) される。当該判別は、画像符号化フォーマット変換装置の処理効率を高めるために、MPEG1の各種フラグ (以下、MPEG1フラグと呼ぶ。) の一つ一つに対して予め定められている。なお、A、B、Cの判別の具体的内容については後述する。

【0027】ここで、上記Aと判別されたMPEG1のパラメータは、特に変換処理等されることなく、直接、MPEG4画像符号化ビットストリームを構成する際のパラメータ (MPEG4パラメータと呼ぶ。) として用いる。

【0028】Bと判別されたMPEG1のパラメータのビットは、MPEG1のVLCデコーダ131により可変長符号の復号化がなされた後、MPEG4のVLCエンコーダ135により可変長符号化される。

【0029】Cと判別されたMPEG1のパラメータのビットは、MPEG1のVLCデコーダ133により可変長符号の復号化がなされた後、パラメータ変換器134により、MPEG4に用いるパラメータ (MPEG4のシンプルフファイルに定義されたパラメータ) に変換され、MPEG4のVLCエンコーダ135へ入力する。MPEG4のVLCエンコーダ135では、当該変換により得られたMPEG4パラメータを可変長符号化する。

【0030】上記判別処理部140により、A、B、Cの何れかに判別されて処理等されたMPEG4パラメータは、バッファ136に格納される。

【0031】MPEG4ビットストリーム構成器137は、バッファ136に格納されたMPEG4パラメータを所望のタイミングに合わせて読み出し、上記ビット読み取り器130からのデータビットと合わせてMPEG4画像符号化ビットストリームを構成する。当該MPEG4画像符号化ビットストリームは、画像符号化フォーマット変換装置123から出力される。

【0032】次に、図3～図9を用い、上記画像符号化フォーマット変換装置にて行われる、MPEG1画像符号化ビットストリームから抽出されたMPEG1パラメータを、MPEG4画像符号化ビットストリームを構成する際に用いるMPEG4パラメータへ変換する流れについて説明する。

【0033】なお、図3～図9は連続した処理を紙面の都合で分割して表しており、また、図3～図9の各図の左側には、入力されたMPEG1画像符号化ビットストリームに対して、MPEG1のビデオシンタックスに従って、ビットの読み取り器130がMPEG1画像符号化ビットストリームから抽出したMPEG1パラメータを示している。また、各パラメータには右向き矢印が付けられており、これら右向き矢印の横には、それぞれ各パラメータが入力したレジスタの番号 (レジスタ群13

9を構成する複数のレジスタのうちの何れかを示す番号)および変換の方法を判別する符号(前記A, B, Cを示す符号)が表示されている。例えば、pel_aspect_ratio→レジスタ3(B)は、MPEG1パラメータpel_aspect_ratioのビットが3番目のレジスタに入力され、さらにBの判別に応じた変換方法に従ってMPEG4パラメータに変換されることを意味している。なお、ビットの読み取り器130にて読み取られたパラメータのうち、MPEG4には使用しないパラメータは、除去される。

【0034】また、図3～図9において、各図の右側には、MPEG4ビットストリーム構成器137により、MPEG4画像符号化ビットストリームを構成するのに必要なMPEG4パラメータを示している。また、各パラメータには左向き矢印が付けられており、これら左向き矢印の横には、MPEG1パラメータが入力したレジスタの番号が表示されている。そのレジスタからMPEG4パラメータを出力する。例えば、aspect_ratio_info←レジスタ3は、MPEG4パラメータaspect_ratio_infoはレジスタ3から読み取ることができることを意味する。レジスタ3では、入力されたMPEG1パラメータpel_aspect_ratioがMPEG4パラメータaspect_ratio_infoに変換されている。

【0035】以下は、レジスタの番号順に沿って、各レジスタに入力されたMPEG1パラメータとそれに対応するMPEG4パラメータ、及び、MPEG1パラメータからMPEG4パラメータへのパラメータ変換について説明する。

【0036】まず、レジスタ1(A)と2(A)には、図10に示すように、horizontal_sizeとvertical_sizeのMPEG1パラメータが入力する。horizontal_sizeは入力画像の横サイズを表し、vertical_sizeは入力画像の縦サイズを表す。上記MPEG1パラメータhorizontal_sizeとvertical_sizeは、そのままMPEG4パラメータvideo_object_layer_widthとvideo_object_layer_heightに代入される。

【0037】レジスタ3(B)には、図11に示すように、pel_aspect_ratioのMPEG1パラメータが入力する。このpel_aspect_ratioは、表示される画像のアスペクト比を表す。当該MPEG1パラメータのpel_aspect_ratioは、MPEG1VLCデコード131によりデコードされた後、MPEG4VLCエンコード132によりエンコードされる。なお、extended PARである場合は、MPEG4には新たなパラメータpar_heightとpar_widthが設定され、par_heightには出力画像の画素のアスペクト比の縦サイズ、par_widthには横サイズの値を入力する。これらpar_heightとpar_widthの比が出力画像のアスペクト比となる。

【0038】レジスタ4(C)には、picture_rateのパラメータが入力する。このパラメータは、図12に示す

ように、変換ツール(パラメータ変換器134)による変換が必要となる。

【0039】ここで、MPEG1の各フレームのデコード時間は、frame_fate_codeとスライスヘッダ(slice header)に格納されているtemporal_referenceにより、計算される。temporal_referenceは、グループオブピクチャ(GOP)の先頭で”0”にリセットされ、フレームがデコードされる度に1増やされる。各フレームのデコード時間はtemporal_referenceの値とframe rateから求められる連続した2フレーム間の時間との積になる。なお、temporal_referenceのパラメータは、フォーマット変換する際に使用しないため除去されている。

【0040】MPEG4画像符号化方式では、各フレームのデコード時間は、vop_time_increment, modulo_time_base及び後述するレジスタ9(B)に入力したtime_codeの各パラメータで決まる。modulo_time_baseは1秒単位でのローカルタイムベースを示す。vop_time_increment_resolutionは、1modulo timebaseを等間隔に分割するサブインターバルの数を示す。vop_time_incrementはmodulo_time_baseで示される同期点から絶対経過時間を示す。

【0041】図13には、それらのパラメータの関係を示す。

【0042】vop_time_incrementは[0, vop_time_increment_resolution)の間の値になっている。vop_time_incrementはvop_time_increment_resolution以上の時間を表すために、modulo_time_baseが用いられる。図13に示しているように、図中の太い枠の中に書かれているのは、modulo_time_baseである。1秒以内は”0”で表し、2秒が経過すると”110”という値になる。n秒が経過すると、”1111・・・10”となる。図中のT1, T2, T3, T4は、各フレームのvop_time_incrementであり、modulo_time_baseから経過した時間を示している。

【0043】レジスタ5(B)と6(B)には、intra_quantizer_matrixとnon_intra_quantizer_matrixが入力する。ここで、MPEG1画像符号化方式には、intra_quantizer_matrixおよびnon_intra_quantizer_matrixという量子化マトリクスのパラメータが規定されている。それらの値は、load_intra_quantizer_matrixおよびload_non_intra_quantizer_matrixというフラグによって決まる。当該フラグが”0”である場合は、MPEG1のシンタックスに定められている値を用いるが、”1”である場合は、ユーザが設定した値が入力される。MPEG4のシンプルプロファイルでは、H. 263と同様の量子化方法を用いるため、intra_quantizer_matrixおよびnon_intra_quantizer_matrixのいずれもない。すなわち、MPEG1とMPEG4の画像符号化方式ではDCT係数の量子化方法が異なる。従って、MPEG4の量子化DCT係数を求めるには、MPEG1の量子化DCT

10

20

30

40

50

T係数を逆量子化し、この逆量子化されたDCT係数をMPEG 4画像符号化方式に従って量子化する必要がある。そのため、MPEG 1の量子化マトリクスをMPEG 1画像符号化方式に従いVLCデコードし、さらにMPEG 1のDCT係数に復号する必要がある。なお、MPEG 4のコアプロファイル (core profile) には、MPEG 1画像符号化方式と同様の量子化方式を用いるため、intra_quantizer_matrixとnon_intra_quantizer_matrixの両方があり、MPEG 1の量子化DCT係数を逆量子化してから、MPEG 4画像符号化方式に従って量子化する必要がなくなる。

【0044】レジスタ7 (A) と8 (A) には、図14に示すように、user_data_start_codeとuser_dataのパラメータが入力する。なお、user_dataはあってもなくてもよいパラメータである。そのため、これらパラメータは変換せずに、そのままMPEG 4画像符号化ビットストリームの一部として用いてもよいし、或いは、除去してもよい。

【0045】レジスタ9 (B) には、図15に示すように、time_codeのパラメータが入力する。なお、当該MPEG 1パラメータのtime_codeのビットの一部は、そのままMPEG 4画像符号化ビットストリームに用いられる。

【0046】レジスタ10 (A) と11 (A) には、図16に示すように、closed_govとBroken_linkのパラメータが入力する。closed_govはMPEG 4画像符号化方式ではGOVヘッダ後のI-VOPに引き続く、B-VOPがバックワード (backward) 予測がイントラ符号化される場合にセットされるフラグである。Broken_linkはMPEG 4画像符号化方式ではGOVヘッダ後のI-VOPに引き続くB-VOPが正しくデコードできない可能性がある場合にセットされるフラグである。MPEG 1画像符号化方式とMPEG 4画像符号化方式では、同様の動作がなされるため、入力されたMPEG 1パラメータは変換せずに、そのままMPEG 4画像符号化ビットストリームに用いられる。

【0047】レジスタ12 (A) と13 (A) には、図17に示すように、user_data_start_codeとuser_dataのパラメータが入力する。user_dataはあってもなくてもよいパラメータである。そのため、これらパラメータは変換せずに、そのままMPEG 4画像符号化ビットストリームの一部として用いてもよいし、或いは除去してもよい。

【0048】レジスタ14 (B) には、図18に示すように、picture_coding_typeのパラメータが入力する。このパラメータは、MPEG 1VLCデコード131によりデコードされ、さらにMPEG 4VLCエンコード132によりエンコードする必要がある。

【0049】レジスタ15 (A) には、図19に示すように、full_pel_forward_vectorのパラメータが入力す

る。MPEG 1画像符号化方式によれば、full_pel_forward_vectorというフラグが” 0” である場合は、半画素精度で動き補償を行うが、” 1” である場合は、整数画素精度で動き補償を行うことになる。また、半画素精度で動き補償を行う際の動きベクトルの値は、整数画素精度で動き補償を行った場合の倍になる。一方、MPEG 4では、半画素精度でしか動き補償を行わない。そのため、full_pel_forward_vectorが” 0” である場合、すなわち、MPEG 1画像符号化方式では半画素精度で動き補償を行う場合、入力されたMPEG 1の動きベクトルはそのままMPEG 4画像符号化の際に用いることができるが、full_pel_forward_vectorが” 1” である場合、すなわちMPEG 1画像符号化方式では整数画素精度で動き補償を行う際、MPEG 4の半画素精度に合わせるため、MPEG 4の動きベクトルはMPEG 1の動きベクトルの2倍にする必要がある。

【0050】レジスタ16 (A) には、図20に示すように、forward_f_codeのパラメータが入力する。当該パラメータは変換せずに、そのままMPEG 4画像符号化ビットストリームに用いる。

【0051】レジスタ17 (A) と18 (A) には、図21に示すように、user_data_start_codeとuser_codeのパラメータが入力する。user_dataはあってもなくてもよいパラメータである。そのため、これらパラメータは変換せずに、そのままMPEG 4画像符号化ビットストリームの一部として用いてもよいし、除去してもよい。

【0052】レジスタ19 (B) には、quantizer_scaleのパラメータが入力する。当該レジスタ19に入力しているquantizer_scaleは、MPEG 1のスライスヘッダ (slice header) に格納されている量子化スケールである。MPEG 1の量子化方法とMPEG 4のシンプルプロファイルの量子化方法とは異なるため、MPEG 4の量子化DCT係数を求める際に、MPEG 1の量子化DCT係数を逆量子化する必要がある。そのため、量子化スケールのパラメータquantizer_scaleをVLCデコードして、値に復号する必要がある。

【0053】レジスタ20 (A) には、macroblock_stuffingのパラメータが入力する。当該macroblock_stuffingは、あってもなくてもよい。そのため、そのままMPEG 4画像符号化ビットストリームに用いてもよいし、除去してもよい。レジスタ21 (B) には、macroblock_typeのパラメータが入力する。このmacroblock_typeはマクロブロックのタイプを意味し、MPEG 4におけるmbpcを設定する際に用いる。このパラメータについては、MPEG 1のVLCデコードを行う必要がある。

【0054】また、レジスタ19 (B) , 21 (B) , 22 (A) , 27 (A) には、quantizer_scale(s)、macroblock_type、quantizer_scale(M)、coded_block_patternのパラメータが入力するが、これら四つのMPEG

10

20

30

40

50

1 パラメータにより、MPEG 4のdquant, mcbpcとcbpyの各パラメータが決まる。

【0055】 先ず、各マクロブロックのdquantのパラメータの求め方を図22を用いて説明する。

【0056】 図22の(a)はMPEG 1のマクロブロックのみのビットストリームの構成を示している。MPEG 1では、マクロブロック毎にquantizer_scaleのパラメータが記述され、図22の(a)では、当該quantizer_scaleのパラメータをq1, q2, q3, ...として表している。また、図22の(b)及び(c)は、
10 図22の(a)の各マクロブロックに対応するMPEG 4のマクロブロックのビットストリームBS1, BS2の構成を示している。MPEG 4では、マクロブロック毎に、dquant、すなわち対象となるマクロブロックの量子化係数と前のマクロブロックの量子化係数の差分が記述されている。MPEG 4画像符号化方式において、dquantは、1, -1, 2, -2のいずれかの値を用いることになっている。そのため、図23に示すように、連続した2つのマクロブロックのquantizer_scaleの差分を
20 MPEG 4のシンタックスの制約に従って3つのケースに分け、dquantを求める処理を行う。

【0057】 図23に示すcase1では、上記差分の値が1, -1, 2, -2の何れかいずれかとなっている場合、当該差分値をそのままdquantとして用いることができる。図22の(b)は全ての差分がcase1に満たす場合である。

【0058】 case2では、差分の値が0、すなわちマクロブロックのquantizer_scaleが送られていない場合、dquantは0にすることができないが、そのマクロブロックのmbtypeにあるdquantのフラグを0にセットすること
30 により、dquantの値を送らず、前のマクロブロックと同様な量子化係数を用いることができる。

【0059】 case3では、差分の絶対値が2以上になった場合に、対象となるマクロブロックの前にリシンクマーカ(resync marker)を配置する。リシンクマーカは、本来、MPEG 4画像復号化方式での復号時にエラー伝播を阻止するために、特定のビットパターンで構成される固定長符号であって、それを検出すれば、次の情報が正しく復号できる。そこで、正しく復号化できるように、ヘッダ情報の一部としてquant_scaleとmacroblock_numberがリシンクマーカの次に記述されている。quant_scaleを設定することによって、連続した2つのマクロブロックの量子化係数の差分を1, -1, 2, -2にすることができる。例えば、図22の(a)のMPEG 1のマクロブロックMB5の量子化係数q5とマクロブロックMB4の量子化係数q4との差分の絶対値が2以上になったとする。その際、図22の(c)に示しているように、MPEG 4のマクロブロックMB5の前にリシンクマーカを配置する。リシンクマーカの次に記述されるquant_scaleをq5+1, q5-1, q5+2, q
50

5-2の何れかにすれば、MPEG 4のマクロブロックMB5のdquant5は、量子化係数q5からquant_scaleを引いた値となるため、-1, 1, -2, 2の何れかになる。

【0060】 以上は、MPEG 1の連続した2つのマクロブロックの量子化係数の差分からdquantを求める方法を説明したが、MPEG 1のビットストリームのデータ構造とMPEG 4のビットストリームの構造が異なるため、階層のヘッダへの処理によるヘッダ内に含まれている量子化係数が変換されることがあり、この場合、ヘッダの先頭のマクロブロックのdquantも新たに求めることがある。

【0061】 ここで、MPEG 1は、シーケンス(sequence)、グループ・オブ・ピクチャ(GOP)、ピクチャ(picture)、スライス(slice)、マクロブロック(MB)、ブロック(block)の6層の階層構造からなっている。それに対して、MPEG 4のシンプルプロファイルの階層構造は、VS (visual sequence)、GOV (Group of Video Object plane)、VOP (Video Object Plane)、マクロブロック(MB)、ブロック(block)の5つになっている。MPEG 4には、MPEG 1のスライス層に相当する階層はないが、VOPの下
の階層として、エラー耐性などで用いられるリシンクマーカと、次の情報からなるビデオパケット(Video Packet)という符号化単位がある。なお、リシンクマーカはMPEG 4のビットストリームを構成するのに必須ではない。

【0062】 従って、MPEG 1からMPEG 4へのフォーマット変換を行う際、MPEG 1のスライス層を除去してもよく、除去した場所でリシンクマーカを配置してもよい。なお、MPEG 4には、vopヘッダ(vop header)の次にリシンクマーカを配置してはいけないことになっている。そのため、MPEG 1のピクチャの先頭のスライスヘッダ(slice header)と先頭でないスライスヘッダへの処理が異なり、先頭のスライスヘッダを除去する処理のみとなる。

【0063】 以下に、それぞれの状況における処理について説明する。

【0064】 先ず、図24を用いてピクチャの先頭のスライスヘッダを除去する処理について説明する。

【0065】 図24の(a)は、MPEG 1のビットストリームの一部を示し、図24の(b)は、変換されたMPEG 4のビットストリームの一部を示している。図24の(a)に示すMPEG 1のビットストリームでは、ピクチャヘッダ(pictureheader)の次にスライスヘッダがあり、当該スライスヘッダに量子化係数quantizer_scaleが記述されている。スライスヘッダの次は、マクロブロックの情報が続いている。また、図24の(b)に示すMPEG 4のビットストリームでは、vopヘッダの次にすぐマクロブロックMBの情報が続いて

おり、vopヘッダにはvop_quantが記述されている。

【0066】ここで、MPEG1のマクロブロックMB1の量子化係数 q_1 が送られている場合において、スライスヘッダに記述されているquantizer_scaleの差が1, -1, 2, -2の何れかであれば、MPEG4のvop_quantの値をMPEG1のquantizer_scaleの値にする。すなわち、 $q_1 - \text{quantizer_scale} = 1, -1, 2, -2$ の何れかのときは、 $\text{vop_quant} = \text{quantizer_scale}$ となる。

【0067】また、MPEG1のマクロブロックMB1の量子化係数 q_1 が送られている場合において、MPEG1の q_1 とquantizer_scaleの差の絶対値が2以上であるときには、MPEG4のvop_quantが $q_1 - 1, q_1 + 1, q_1 - 2, q_1 + 2$ の何れかになされる。すなわち、 $|q_1 - \text{quantizer_scale}| > 2$ のときは、 $\text{vop_quant} = q_1 - 1, q_1 + 1, q_1 - 2, q_1 + 2$ の何れかとなる。その際、dquant1はそれぞれ1, -1, 2, -2になる。

【0068】また、MPEG1のマクロブロックMB1の量子化係数 q_1 が送られている場合において、MPEG1の q_1 とquantizer_scaleの差が0であるときには、MPEG4のvop_quantはMPEG1のquantizer_scaleと同様な値を取る。すなわち、 $q_1 - \text{quantizer_scale} = 0$ のときは、 $\text{vop_quant} = \text{quantizer_scale}$ となる。その際、マクロブロックMB1のmbtypeにあるdquantフラグは0に設定される。

【0069】一方、MPEG1のマクロブロックMB1の量子化係数 q_1 が記述されていない場合、MPEG1のマクロブロックMB1の量子化係数 q_1 は、quantizer_scaleと同様な値になるため、MPEG4のvop_quantはquantizer_scale-1, quantizer_scale+1, quantizer_scale-2, quantizer_scale+2の何れかになされる。すなわち、 $\text{vop_quant} = \text{quantizer_scale} - 1, \text{vop_quant} = \text{quantizer_scale} + 1, \text{vop_quant} = \text{quantizer_scale} - 2, \text{vop_quant} = \text{quantizer_scale} + 2$ の何れかとなされる。

【0070】次に、図25を用いて、ピクチャの先頭以外のスライス層の処理について説明する。

【0071】図25の(a)はMPEG1のビットストリームの一部を示し、図25の(b)及び(c)は、変換されたMPEG4のビットストリームの一部を示している。図25の(a)に示すように、MPEG1のビットストリームのスライスヘッダはピクチャの先頭ではない。

【0072】ピクチャの先頭以外のスライス層については、MPEG4への変換の際に、除去しても良く、また、除去したところでリシンクマーカを配置してもよい。

【0073】図25の(b)に示すMPEG4のビットストリームは、MPEG1のビットストリームに存在し

ていたスライスヘッダが除去された例を示している。この図25の(b)のように、スライスヘッダを除去できる条件は、MPEG1のマクロブロックMB1'の量子化係数 q_1' とマクロブロックMB6の量子化係数 q_6 の差分が1, -1, 2, -2のいずれかの場合である。

【0074】また、MPEG1のマクロブロックMB1'の量子化係数 q_1' と q_6 の差分の絶対値が2以上になった場合は、図25の(c)に示しているように、スライスヘッダの代わりにリシンクマーカを配置する。

【0075】例えば、MPEG1のマクロブロックMB1'の量子化係数 q_1' が送られている場合において、MPEG1の q_1' とスライスヘッダに記述されているquantizer_scaleとの差が1, -1, 2, -2の何れかであるときには、MPEG4のquant_scaleをMPEG1のquantizer_scaleにする。すなわち、 $q_1' - \text{quantizer_scale} = 1, -1, 2, -2$ の何れかのときは、 $\text{quant_scale} = \text{quantizer_scale}$ となる。

【0076】また、MPEG1のマクロブロックMB1'の量子化係数 q_1' が送られている場合において、MPEG1の q_1' とquantizer_scaleの差分の絶対値が2以上になったときには、MPEG4のquant_scaleがMPEG1の $q_1' - 1, q_1' + 1, q_1' - 2, q_1' + 2$ の何れかになされる。その際、MPEG4のdquant1'はそれぞれ1, -1, 2, -2になる。すなわち、 $|q_1' - \text{quantizer_scale}| > 2$ のときは、 $\text{quant_scale} = q_1' - 1, q_1' + 1, q_1' - 2, q_1' + 2$ の何れかとなる。

【0077】また、MPEG1のマクロブロックMB1'の量子化係数 q_1' が送られている場合において、MPEG1の q_1' とquantizer_scaleの差が0であるときには、MPEG4のquant_scaleはMPEG1のquantizer_scaleと同様な値を取る。すなわち、 $q_1' - \text{quantizer_scale} = 0$ のときは、 $\text{quant_scale} = \text{quantizer_scale}$ となる。その際、マクロブロックMB1'のmbtypeにあるdquantフラグは0に設定される。

【0078】一方、MPEG1のマクロブロックMB1'の量子化係数 q_1' が記述されていない場合、MPEG4のquant_scaleはquantizer_scale-1, quantizer_scale+1, quantizer_scale-2, quantizer_scale+2の何れかになされる。すなわち、 $\text{quant_scale} = \text{quantizer_scale} - 1, \text{vop_quant} = \text{quantizer_scale} + 1, \text{vop_quant} = \text{quantizer_scale} - 2, \text{vop_quant} = \text{quantizer_scale} + 2$ の何れかとなされる。

【0079】図3～図9の説明に戻り、レジスタ21(B)に入力したmacroblock_typeはMPEG1VLCデコーダ131によりデコードされ、さらにMPEG4VLCエンコーダ132によりエンコードされ、図26に示すように、MPEG4のパラメータmbtypeに変換される。

【0080】また、レジスタ27(C)に入力したcode

d_block_patternは、マクロブロックの4つの輝度ブロックと2つの色差ブロックの6個のブロックのpattern_codeを決めている。pattern_codeは1であればブロックはcoded、0であればブロックはnot_codedである。図27に示すように、pattern_code[0~3]は4つの輝度ブロック、pattern[4, 5]は2つの色差ブロックのcoded, not_codedの情報を示している。求められたdquantとmbtype及びcoded_block_patternにより、MPEG4のmcbpcが決められる。また、cbpyはcoded_block_patternにより決まる。なお、MPEG1のスキップマクロブロックは、MPEG4のnot_codedのフラグが1の時に相当する。最後に、求められたcbpyとmcbpcは、MPEG4 V L Cエンコーダ135で符号化される。

【0081】レジスタ23 (C), 24 (C), 25 (C), 26 (C) には、図28に示すように、motion_horizontal_forward_code, motion_horizontal_forward_r, motion_vertical_forward_code, motion_vertical_forward_rのパラメータが入力する。これらパラメータにより、MPEG1とMPEG4の動きベクトルの予測誤差を符号化する際、用いる動きベクトルMVの予測値が異なる。

【0082】図29を用いて、MPEG1とMPEG4の動きベクトルMVの予測方法を説明する。

【0083】図29の(a)は、MPEG1のマクロブロックMBの動きベクトルMVを予測する際、予測値は、直前のマクロブロックMB1の動きベクトルMV1を用いることを示している。

【0084】図29の(b)は、MPEG4のマクロブロックMBの動きベクトルMVを予測する際、用いる予測値を示している。画像符号化フォーマット変換する際、MPEG4には、4つの8画素×8画素のブロックの動きベクトルMV_{8x8}を用いると、処理が煩雑になる。そのため、8画素×8画素のブロックの動きベクトルMV_{8x8}による動き補償は行わないで、一つのマクロブロックに対して16画素×16画素のマクロブロックの動きベクトルMV_{16x16}を用いる。図29の(b)には、ラインline nとラインline n+1の2ラインのマクロブロックが格納されている。対象となるマクロブロックは、ラインline n+1にある。予測値は、近傍のマクロブロックMB1の動きベクトルMV1、マクロブロックMB2の動きベクトルMV2、マクロブロックMB3の動きベクトルMV3を用いる。マクロブロックMBの動きベクトルMVの予測値は、同じラインの左にあるマクロブロックMB1の動きベクトルMV1、ラインをまたがって上のラインにあるマクロブロックMB2の動きベクトルMV2と、右上のマクロブロックMB3の動きベクトルMV3の中間値となる。

【0085】従って、レジスタ23~26に入力したMPEG1のパラメータは、先ずMPEG1のV L Cデコーダ133により復号化され、MPEG1の動きベクトルMVの予測値と差分により動きベクトルMVの値が復号される。復号された動きベクトルMVは、MPEG4の予測方法に従って予測値が選択し、予測誤差を求め、MPEG4 V L Cエンコーダ135によりエンコードすることが必要である。

【0086】レジスタ28 (C), 29 (C), 30 (C), 31 (C) には、図30に示すように、dct_dc_size_luminance, dct_dc_differential, dct_dc_size_chrominance, dct_dc_differentialのパラメータが入力する。これらパラメータにより、イントラマクロブロックのDC係数の予測誤差を求める際に用いるDC係数の予測値は、MPEG1とMPEG4とで異なる。図31を用いて、MPEG1とMPEG4のイントラマクロブロックのDC係数の予測方法を説明する。

【0087】図31の(a)は、MPEG1のDC係数の予測を表している。図中の太い枠で囲まれていたマクロブロックは、さらに細い点線で24のブロックに区切られる。ブロックは符号化される順で、1, 2, 3, 4のように番号がつけられている。ブロックの左上にある斜線で示した小さいブロックは、DC係数を意味する。矢印は予測の方向を示している。矢印の始点にあるDC係数は、矢印の終点にあるDC係数の予測値である。なお、DC係数の予測を行うのはイントラマクロブロックのみである。DC予測値は、スライス先頭および、非イントラマクロブロックのとき1024にリセットされる。ここでは、輝度のイントラマクロブロックのDC係数の予測方法を例として挙げるが、色差のイントラマクロブロックのDC係数に対しても同様の処理を行う。

【0088】先ず、dct_dc_size_luminance及びdct_dc_differentialと、dct_dc_size_chrominance及びdct_dc_differentialを、MPEG1 V L Cデコーダ133で復号し、予測誤差をmp1_Δnとし、DC係数をmp1_dcnにする。なお、nはブロックの番号を示している。また、MPEG1 V L Cデコード133でデコードされた量子化係数をmp1_qnにすると、ブロック1のDC予測誤差がmp1_Δ1, DC係数がmp1_dc1, 量子化係数がmp1_q1となる。以下同様に、ブロック2のDC予測誤差がmp1_Δ2, DC係数がmp1_dc2, 量子化係数がmp1_q2となるようにし、ブロックnのDC予測誤差がmp1_Δn, DC係数がmp1_dcn, 量子化係数がmp1_qnとなるようにする。各ブロックのDC係数mp1_dcnは、以下の式によりDC係数に復号される。

【0089】

$$\text{ブロック 1 : mp_dc1} = 1024 + \text{mp1_}\Delta 1 * \text{mp1_q1} \quad (1)$$

$$\text{ブロック 2 : mp_dc2} = \text{mp1_dc1} + \text{mp1_}\Delta 2 * \text{mp1_q2} \quad (2)$$

$$\text{ブロック 3 : mp_dc3} = \text{mp1_dc2} + \text{mp1_}\Delta 3 * \text{mp1_q3} \quad (3)$$

$$\text{ブロック 4 : mp_dc4} = \text{mpl_dc3} + \text{mpl_}\Delta 4 * \text{mpl_q4} \quad (4)$$

・
・

また、ブロック 13 からは、再び予測値が 1024 になる。

したがって、以下の式により DC 係数に復号され **【0090】**

$$\text{ブロック 13 : mp_dc13} = 1024 + \text{mpl_}\Delta 13 * \text{mpl_q13} \quad (5)$$

$$\text{ブロック 14 : mp_dc14} = \text{mpl_dc13} + \text{mpl_}\Delta 14 * \text{mpl_q14} \quad (6)$$

・
・

$$\text{ブロック n : mp_dcn} = \text{mpl_dc(n-1)} + \text{mpl_}\Delta n * \text{mpl_qn} \quad (7)$$

このように、MPEG 1 のブロックの DC 係数は、逆量子化され DC 係数の予測誤差と、前のブロックの復号された DC 係数との和になる。

【0091】 次に、図 31 の (b) は、MPEG 4 のシンタックスで定められている予測方法を示している。

【0092】 ここで、ブロック A, B, X, Y, Z, W は同じラインに格納されているが、ブロック C, D, E は上のラインにあるブロックである。例えば、ブロック X の DC 係数を求める際、隣接するブロック B と上のラインにあるブロック C, D の DC 係数を比較することにより、ブロック B あるいはブロック D を予測に用いることが決まる。

【0093】 $F[0][0]$ を DC 係数にするとすると、ブロック X の DC 係数の予測値を求める際には、 $|F_B[0][0] - F_C[0][0]| < |F_C[0][0] - F_D[0][0]|$ のときブロック D を予測に用い、そうでない場合はブロック B を予測に用いる。

【0094】 また、ブロック Y の DC 係数の予測値を求める際には、 $|F_X[0][0] - F_D[0][0]| < |F_D[0][0] - F_E[0][0]|$ のときブロック E を予測に用い、そうでない場合はブロック X を予測に用いる。

【0095】 ブロック Z の DC 係数の予測値を求める際には、 $|F_A[0][0] - F_B[0][0]| < |F_B[0][0] - F_X[0][0]|$ のときブロック X を予測に用い、そうでない場合はブロック A を予測に用いる。

【0096】 ブロック W の DC 係数の予測値を求める際には、 $|F_Z[0][0] - F_X[0][0]| < |F_X[0][0] -$

$$\text{dct_recon}[m][n] = (2 * \text{dct_zz}[i] * \text{quantizer_scale} * \text{intra_quant}[m][n] / 16 \quad (8)$$

インターの DCT 係数の逆量子化は式 (9) に示すよう **【0100】**

になされる。

40

$$\begin{aligned} \text{dct_recon}[m][n] = & ((2 * \text{dct_zz}[i]) + \text{sign}(\text{dct_zz}[i])) \\ & * \text{quantizer_scale} * \text{non_intra_quant}[m][n] / 16 \end{aligned} \quad (9)$$

MPEG 4 では、VOP 毎に設定する vop_quant とマクロブロック間の量子化スケールの差分 dquant から求められたマクロブロック毎の quantizer_scale によって量子化を行っている。式 (10)、式 (11) は、MPEG

$$\begin{aligned} & \text{dct_zz}[i] \text{ が偶数の場合、} \\ & \text{dct_recom}[m][n] = (2 * \text{dct_zz}[i] + \text{sign}(\text{dct_zz}[i])) * \text{quantize_scale} \end{aligned} \quad (10)$$

$$\begin{aligned} & \text{dct_zz}[i] \text{ が奇数の場合、} \\ & \text{dct_recom}[m][n] = (2 * \text{dct_zz}[i] + \text{sign}(\text{dct_zz}[i])) \end{aligned}$$

$F_Y[0][0]$ のときブロック Y を予測に用い、そうでない場合はブロック Z を予測に用いる。

【0097】 上述したように、MPEG 4 は、隣接するブロックの DC 係数の比較により予測値が決まる。各ブロックの $\text{dct_dc_size_luminance}$ 及び $\text{dct_dc_differential}$ と、 $\text{dct_dc_size_chrominance}$ 及び $\text{dct_dc_differential}$ は、MPEG 1 VLC デコーダ 133 で復号され、さらにレジスタ 5, 6 に入力された量子化マトリクスおよびレジスタ 19, 22 に入力された量子化スケールを用いて、式 (1) ~ (7) に従って、MPEG 1 の DC 係数に復号される。復号された DC 係数は、MPEG 4 の予測値を求める方法に従って、DC 係数の予測値が決定され、DC 係数め予測誤差が求められる。DC 係数の予測誤差は、MPEG 4 VLC エンコーダ 135 によりエンコードされる。

【0098】 レジスタ 32 (C)、33 (C) には、 dct_coef_first 、 dct_coef_next のパラメータが入力する。イントラ AC 係数及びインターの DCT 係数の量子化方法は、MPEG 1 と MPEG 4 のシンプルプロファイルとで異なる。MPEG 1 は、スライスヘッダとマクロブロック毎に設定可能な quantizer_scale 及びシーケンス毎に設定する $\text{intra_quant_matrix}$ 、 $\text{non_intra_quant_matrix}$ によって、DCT 係数の量子化を行っている。イントラ AC 係数の逆量子化は、式 (8) に示すようになされる。

【0099】

4 のイントラ AC 係数及びインター DC 係数逆量子化方式である。

【0101】

*quantize_scale=sign(dct_zz[i])

(11)

なお、dct_recon[m][n]は、復元されたDCT係数であり、dct_zz[i]はジグザグスキャンした2次元係数を1次元にした値である。

【0102】従って、量子化DCT係数の変換をするには、dct_coef_firstとdct_coef_nextをMPEG1VLCデコーダ133よりdct_zz[i]に復号し、逆量子化によりdct_recon[m][n]に復号する。復号されたdct_recon[m][n]をMPEG4のdct_zz[i]を符号化する際、用いる可変長符号化方法が異なる。MPEG1は、2次元のラン、レベルペアを用いるが、MPEG4は、3次元のラン(run)、レベル(level)、ラスト(last)ペアを用いる。その関係は、図32のようになっている。

【0103】その後、求められたラスト、ラン、レベルペアは、MPEG4VLCエンコーダ135によりエンコードされる。

【0104】レジスタ34(B)、35(B)には、macroblock_escape、macroblock_address_incrementのパラメータが入力する。macroblock_address_incrementは33以上になった時、macroblock_escapeを用いる。macroblock_escapeの値は33である。スキップマクロブロックの数は、macroblock_escape * n + macroblock_address_incrementになる。なお、nはmacroblock_escapeが表れた回数を意味する。

【0105】レジスタ36(A)、37(A)、38(A)、39(A)、40(A)、41(A)、42(A)には、macroblock_type、quantizer_scale(M)、motion_horizontan_forward_code、motion_horizontal_forward_r、motion_vertical_forward_code、motion_vertical_forward_r、coded_block_patternの各パラメータが入力する。これらパラメータについては、先頭のマクロブロックのパラメータ変換と同様な処理がなされる。

【0106】以上の42個のレジスタにMPEG1のパラメータを入力し、そこでMPEG4のパラメータが生成される。なお、それ以外では、MPEG4のリシンクマーカの packets ヘッダ (packet header) にマクロブロックナンバ (macroblocknumber) を設定するフラグがあるため、マクロブロックのナンバをカウントするカウンタを格納するバッファも必要である。

【0107】次に、本発明の第2の実施の形態について説明する。

【0108】第2の実施の形態の構成は、前記第1の実施の形態の構成と同様であるが、画像符号化フォーマット変換器の一部の動作が異なる。

【0109】前述の図29、図31と同様に表す図33と図34を用いてその動作を説明する。

【0110】第2の実施の形態の場合の画像符号化フォーマット変換器内では、MPEG4のビットストリームを構成する際に、各行の先頭にリシンクマーカを配置するという動作を加える。

【0111】第1の実施の形態に説明したように、リシンクマーカは、本来、MPEG4復号時エラー伝播を阻止するための同期語で、正しく復号できる開始点を示している。リシンクマーカの次にヘッダ情報の一部が記述されているため、その情報を用いて正しく復号化できる。例えば、行の先頭にリシンクマーカを配置すると、その行に格納されているマクロブロックデータは他の行のマクロブロック情報を参照することなく、独立して符号化される。したがって、動きベクトルの予測及びイントラマクロブロックのDC係数の予測は、行にまたがったマクロブロックを参照することがなくなり、変換の処理両が少なくなる。

【0112】図33を用いて、リシンクマーカを配置した後の動きベクトルの変換について説明する。

【0113】図33の(a)は、MPEG1の動きベクトルの予測方法を示している。マクロブロックMBの動きベクトルMVの予測値は、図の左側のマクロブロックMB1の動きベクトルMV1である。

【0114】図33の(b)は、行の先頭にリシンクマーカを配置した場合、MPEG4の動きベクトルMVの予測方法を示している。図29の(b)と異なるのは、上のラインをまたがったマクロブロックMBの動きベクトルMVを参照することがなくなったことである。そのため、常に左側にあるマクロブロックMB1の動きベクトルMVが予測値となり、MPEG1と同様な予測方法になっている。

【0115】従って、この第2の実施の形態の場合のレジスタ24、25、26および27に入力したMPEG1のパラメータは、MPEG1のVLCデコーダを通してデコードされた後、パラメータ変換せずに、MPEG4のVLCエンコーダでMPEG4で用いるパラメータに変換される。

【0116】それ以外のリシンクマーカを配置した際、動きベクトルMVへの処理は以下になる。

【0117】まず、MPEG1のピクチャ層の先頭以外のスライス層を変換処理した場合、スライスヘッダの先頭のマクロブロックMBは動きベクトル予測誤差が0で、動きベクトル係数を用いる。リシンクマーカの先頭のマクロブロックMBも動きベクトル予測誤差が0で、動きベクトル係数を用いる。従って、スライスヘッダに相当するリシンクマーカを配置した場合、MPEG1とMPEG4は一致する。しかし、リシンクマーカを配置しない場合は、先頭のマクロブロックMBの動きベクトルMVは動きベクトル係数の代わりに、予測誤差で符号化することになる。そのため、前のマクロブロックMBの動きベクトルMVとの差を求めることができるように、動きベクトル係数に復号する必要がある。

【0118】一方、連続した2つのマクロブロックMBのquantizer_scaleの差分が2以上になった場合におい

て、リシンクマーカが配置された場合は、MPEG 1では、動きベクトルMVの予測誤差が符号化されているが、MPEG 4では、リシンクマーカが配置された後、動きベクトル係数を符号化することになる。そのため、リシンクマーカ直後のマクロブロックMBの動きベクトル係数を求めるには、そのマクロブロックMBの前にあるマクロブロックMBの動きベクトル係数を復号する必要がある。

【0119】それ以外のマクロブロックMBの動きベクトルMVは、MPEG 1とMPEG 4とも、動きベクトル予測誤差を用いるため、変換する必要がない。

【0120】図34を用いて、イントラマクロブロックのDC係数の変換について説明する。

【0121】図34の(a)は、MPEG 1でのイントラマクロブロックのDC係数の予測方法を示している。

【0122】図34の(b)は、リシンクマーカを配置

$$\text{ブロック 1 : } mp4_dc1 = 1024 + mp4_Δ1 * mp4_q1 \quad (12)$$

ここで、同じブロックのDC係数が同じ値であると仮定すると、式(13)で表すことができる。

$$mp4_dc1 = mp1_dc1 \quad (13)$$

この式(13)を式(12)に代入すると、式(14)のようになる。

$$mp1_dc1 = 1024 + mp4_Δ1 * mp4_q1 \quad (14)$$

従って、 $mp4_Δ1$ は次式(15)のようになる。

$$mp4_Δ1 = (mp1_dc1 - 1024) / mp4_q1 \quad (15)$$

式(15)の $mp4_Δ1$ が整数なので、実際の $mp4_Δ1$ は、式(16)のようになる。

$$mp4_Δ1' = [mp4_Δ1] = [(mp1_dc1 - 1024) / mp4_q1] \quad (16)$$

なお、式(16)の[]は整数化することを意味する。

1は、式(17)のようになる。

【0129】量子化誤差を考慮すると、実際MPEG 4デコーダで逆量子化されるブロック1のDC係数 $mp4_dc$

$$mp4_dc1 = 1024 + [mp4_Δ1] * mp4_q1 \quad (17)$$

また、ブロック2は、上側にブロックがないため、左側にあるブロック1のDC係数を予測値とする($mp4_dc2 = mp4_dc1 + mp4_Δ2 * mp4_q2$)と同様に、 $mp4_dc$

$$mp4_Δ2' = [mp4_Δ2] = [(mp1_dc2 - mp4_dc1) / mp4_q2] \quad (18)$$

実際、MPEG 4デコーダで量子化されたブロック1のDC係数は、式(19)のようになる。

$$mp4_dc2 = mp4_dc1 + [mp4_Δ2] * mp4_q2 \quad (19)$$

次に、ブロック3は、左側にブロックがないため、上側にあるブロック1のDC成分を予測値とし、上述同様に

$$mp4_Δ3' = [mp4_Δ3] = [(mp1_dc3 - mp4_dc1) / mp4_q3] \quad (20)$$

実際MPEG 4デコーダで量子化されたブロック1のDC係数は、式(21)のようになる。

$$mp4_dc3 = mp4_dc1 + [mp4_Δ3] * mp4_q3 \quad (21)$$

ブロック4は、近傍のブロック1, 2, 3のDC係数を比較することによって、DC係数の予測値が決まる。この時点では、ブロック1, 2, 3のDC係数がすでに計算されているため、 $|mp4_dc3 - mp4_dc1| < |mp4_dc2 -$

した後、MPEG 4のDC係数の予測方法を示している。

【0123】この図34から判るように、またがったラインのマクロブロックMBを参照しないため、ライン内のブロックのDC係数のみを予測値として用いる。ここで、MPEG 1のDC係数はすでに逆量子化され、DC係数に復号されているとする。第1の実施の形態と同様に、図34の(b)のMPEG 4のブロック1のDC予測誤差を $mp4_Δ1$ 、DC係数を $mp4_dc1$ 、量子化係数を $mp4_q1$ 、ブロック2のDC予測誤差を $mp4_Δ2$ 、DC係数を $mp4_dc2$ のように、量子化係数を $mp4_q2$ 、ブロックnのDC予測誤差を $mp4_Δn$ 、DC係数を $mp4_dcn$ 、量子化係数を $mp4_qn$ とすると、各ブロックのDC係数 $mp4_dcn$ およびDC予測誤差 $mp4_Δn$ は、以下の式により求められる。

【0124】

【0125】

【0126】

【0127】

【0128】

【0130】

2= $mp1_dc2$ と仮定すると、式(18)で表される。

【0131】

【0132】

して式(20)で表される。

【0133】

【0134】

$mp4_dc1$ | のとき、ブロック2を予測に用いる。

【0135】同様に、式(22)と式(23)の計算を行う。

【0136】

$$mp4_Δ4' = [mp_Δ4] = [(mp1_dc4 - mp4_dc2) / mp4_q4] \quad (22)$$

$$mp4_dc4 = mp4_dc2 + [mp4_Δ4] * mp4_q4 \quad (23)$$

そうでなければ、式(24)、式(25)のように、ブロック3のDC係数を予測値として用いる。

$$mp4_Δ4' = [mp_Δ4] = [(mp1_dc4 - mp4_dc3) / mp4_q4] \quad (24)$$

$$mp4_dc4 = mp4_dc3 + [mp4_Δ4] * mp4_q4 \quad (25)$$

ブロック13は、再びブロック1と同様に、予測値を1024として計算し、ブロック14、15、16はそれぞれ、ブロック2、3、4と同様な計算をする。ブロック17、18、21、22はそれぞれ前のブロックのDC係数を予測値とする。ブロック19、20、23、24のみは、それぞれ前と上のブロックのいずれのDC係数を予測値として用いる。すなわち、本実施の形態では、すべてのイントラマクロブロックのブロックに対し、近傍のブロックのDC係数を比較することがなくなり、一部のイントラマクロブロックのブロックに対し、近傍のブロックのDC係数を比較し、予測値を決めることとなる。そのため、変換処理が軽くなる。

【0138】なお、色差のイントラマクロブロックのDC係数の予測は、1つのマクロブロックMBに対して、1つの色差ブロックのみがラインに格納されている。そのため、輝度のイントラのDC係数との予測方法が異なり、前の色差ブロックのDC係数を予測値として用いる。すなわち、行の先頭にリシンクマーカを配置した場合、MPEG4のイントラマクロブロックの色差のDC係数の予測方法と、MPEG1とが一致する。

【0139】上述した計算で求めた $mp4_Δn$ は、MPEG4のVLCエンコードにより、 $dct_dc_size_luminance$ 及び $dct_dc_differential$ と、 $dct_dc_size_chrominance$

$$mp4_Δ2 = mp1_Δ1 * mp1_q2 / mp4_q2 = mp1_Δ1 * 8 / mp2_q2 \quad (27)$$

ブロック3の場合、MPEG1ではブロック2のDC係数を予測値として用いるが、MPEG4ではブロック1のDC係数を予測値として用いる。例えば、MPEG1では $mp1_dc3 = mp1_dc2 + mp1_Δ3 * mp1_q3$ 、MPEG4では

$$\begin{aligned} mp4_Δ3 &= (mp1_dc2 - mp4_dc1 + mp1_Δ3 * mp1_q3) / mp4_q3 \\ &= (mp1_dc2 - mp1_dc1 + mp1_Δ3 * mp1_q3) / mp4_q3 \\ &= (mp1_Δ2 * mp1_q2 + mp1_Δ3 * mp1_q3) / mp4_q3 \\ &= (mp1_Δ2 + mp1_Δ3) * 8 / mp4_q3 \end{aligned} \quad (28)$$

ブロック4の場合、MPEG4では近傍のブロック1、2、3のDC係数を比較することによって、DC係数の予測値が決まる。例えば、MPEG1では $mp1_dc4 = mp1_dc3 + mp1_Δ4 * mp1_q4$ 、MPEG4では

$$\begin{aligned} |mp1_dc3 - mp1_dc1| &< |mp1_dc2 - mp1_dc1| \\ |mp1_dc3 - mp1_dc2 + mp1_dc2 - mp1_dc1| &< |mp1_dc2 - mp1_dc1| \\ |mp1_Δ3 + mp1_Δ2| &< |mp1_Δ2| \end{aligned}$$

$mp1_Δ2 > 0$ 、 $mp1_Δ3 < 0$ 、若しくは、 $mp1_Δ2 < 0$ 、 $mp1_Δ3 > 0$

つまり、 $mp1_Δ2 * mp1_Δ3 < 0$ であれば、ブロック2のDC係数を予測値とする。

【0148】また、 $mp4_dc4 = mp4_dc2 + mp4_Δ4 * mp4_q4$

e及び $dct_dc_size_differential$ になされる。

【0140】以上のイントラのDC係数は、量子化誤差を考慮して計算されているため、値は正確に求められるが、処理量が大い。そこで、量子化誤差を考慮せず、MPEG1とMPEG4のDC係数が同様な値にすると仮定して、量子化DC係数を求めると、処理量がさらに少なくなる。

【0141】図34の(b)と同様な予測方法である。

【0142】MPEG1のDC係数の各量子化係数は $mp1_q1 = mp1_q2 = \dots = mp1_qn = 8$ である。

【0143】ブロック1はMPEG1とMPEG4とも同様な予測値を用いる。例えば、MPEG1では $mp1_dc1 = 1024 + mp1_Δ1$ 、MPEG4では $mp4_dc1 = 1024 + mp4_Δ1$ 、 $mp1_dc1 = mp4_dc1$ と仮定すると、式(26)のようになる。

$$【0144】 mp4_Δ1 = mp1_Δ1 \quad (26)$$

また、ブロック2もMPEG1とMPEG4が同様な予測値を用いる。例えば、MPEG1では $mp1_dc2 = mp1_dc1 + mp1_Δ2 * mp1_q2$ 、MPEG4では $mp4_dc2 = mp4_dc1 + mp4_Δ2 * mp4_q2$ 、 $mp1_dc2 = mp4_dc2$ と仮定すると、式(27)のようになる。

【0145】

は $mp4_dc3 = mp4_dc1 + mp4_Δ3 * mp4_q3$ 、 $mp1_dc3 = mp4_dc3$ と仮定すると、式(28)のようになる。

【0146】

$|mp1_dc1| < |mp4_dc2 - mp4_dc1|$ であれば、ブロック2を予測に用いる。すなわち以下のようになる。

【0147】

4、 $mp4_dc4 = mp1_dc4$ と仮定すると、同様に式(29)になる。

【0149】

$$mp4_Δ4 = (mp1_Δ3 + mp1_Δ4) * 8 / mp4_q4 \quad (29)$$

逆に、 $mp1_Δ2 * mp1_Δ3 > 0$ であれば、ブロック3の

DC係数を予測値とする。

【0150】 $mp4_dc4 = mp4_dc3 + mp4_Δ4 * mp4_q4$

同様に式(30)になる。

【0151】

$mp4_Δ4 = mp1_Δ4 * 8 / mp4_q4$ (30)

なお、 $mp1_Δ2 * mp_Δ3$ の符号は、 $mp1_Δ2$ と $mp_Δ3$ の可変長符号によって求めることができる。MPEG1では、 $mp1_Δ2$ 又は $mp1_Δ3$ が正数であれば、1番最後のビットは0となる。逆に、負数であれば1となる。従って、最後のビットの論理積を取ることにによって、 $mp1_Δ2 * mp1_Δ3$ の符号がわかる。すなわち、近傍のブロックのDC係数に復号せずに、可変符号のままに比較し、予測値を決めることができる。

【0152】ブロック13からの計算もブロック1～4までと同様にできる。

【0153】また、MPEG4のDC係数の量子化係数 $mp4_qn$ は、AC係数の量子化係数によって、以下のような処理が行われている。なお、図35に示すように、AC係数の量子化係数 $quantizer_scale$ が1～4の場合は、MPEG4のDC係数の量子化係数はMPEG1と同様に8になる。

【0154】式(26)～式(29)に示した $mp4_qn$ を用いた $8/mp4_qn$ は、論理回路で簡単に演算できるように、図36に示しているような近似式により求める。なお、図36には、AC係数の量子化係数が5～8の値を例にしているが、他の値に対しても同様な近似式が求められる。

【0155】従って、MPEG4の量子化DC係数は、すべてMPEG1の量子化DC係数より求めることができる。また、予測を求める際、近傍のブロックのDC係数の比較もMPEG1の量子化DC係数の可変符号を用いて行うことができる。そのため、MPEG1の量子化DC係数をMPEG1で逆量子化し、DC係数の復号を行ってからMPEG4で量子化することがなくなり、処理が少なくなる。

【0156】次に、本発明の第3の実施の形態について説明する。

【0157】本発明の第3の実施の形態の構成は、基本的には図1に示した第1の実施の形態の構成と同様であるが、第3の実施の形態の場合のMPEG1画像符号化器120では、MPEG1のパラメータであるウェイトイングマトリクス(weightinb matrix)を16に固定する。その結果、MPEG4のシンプルプロファイルに用いる量子化方法はMPEG1と一致し、MPEG1の逆量子化を行った後、MPEG4の量子化をする必要がなくなり、処理が軽くなる。

【0158】すなわち、MPEG1には、シーケンスに $intra_quantizer_matrix$ と $non_intra_quantizer_matrix$ のような量子化マトリクスがあるが、MPEG4のシンプルプロファイルにはない。図1に示した第1の実施の

形態のフォーマット変換では、量子化DC係数をMPEG1で逆量子化してDC係数に復元し、それをMPEG4で量子化をする必要があるため、計算量が多い。これに対して、第3の実施の形態では、計算量を減らすため、入力された画像信号をMPEG1画像符号化器120により符号化する際に、MPEG1のパラメータである $intra_quantizer_matrix$ の64個の値をすべて16に設定する。そうすることによって、MPEG1の逆量子化方式とMPEG4の逆量子化方式が一致し、MPEG4の量子化DC係数を求める際に、MPEG1の逆量子化をしてからMPEG4の量子化する必要がなくなる。なお、この第3の実施の形態の場合、MPEG1の量子化パラメータである $intra_quantizer_matrix$ 、 $non_intra_quantizer_matrix$ はレジスタ5、6に入力せず、除去する。

【0159】次に、本発明の第4の実施の形態について説明する。

【0160】本発明の第4の実施の形態の構成は、基本的には図1に示した第1の実施の形態の構成と同様であるが、第4の実施の形態の場合のMPEG1画像符号化器120では、輝度の動きベクトルへの処理が加わる。すなわち、輝度の動きベクトルMVとして例えば1, 5, 9, ...のような4で割って余りが1となる値が検出された場合、1を減らした値を符号化時に用いる輝度の動きベクトルMVにする。つまり、クロマの動きベクトルの求め方はMPEG1とMPEG4が異なるため、動き補償する際に、ドリフト誤差が生じ、画質が劣化するので、MPEG4のクロマの動きベクトルをMPEG1のクロマの動きベクトルに一致させることにより、画質を向上させている。

【0161】より具体的に説明すると、輝度の動きベクトルMVに対して、クロマの動きベクトルは、MPEG1では輝度の動きベクトルMVの二分の一になっているが、MPEG4ではMPEG4のシンタックスに従った求め方がある。例えば、図37の(a)は、輝度の動きベクトルMVが0, 1, 2, 3, ...のようになっていたときのMPEG1とMPEG4のクロマの動きベクトルMVを示しており、図中円で囲まれた数字で表される動きベクトルの場合、一つの輝度の動きベクトルMVに対して、MPEG1とMPEG4のクロマの動きベクトルMVが異なる。それ以外は、MPEG1とMPEG4のクロマの動きベクトルMVが常に一致している。このように、クロマの動きベクトルMVが異なると、動き補償を行う際に、クロマの値が少しずれてしまう。この場合、イントラVOPがリセットされない限りは、クロマの誤差が蓄積して、画像の色が部分的に緑になる。そういう劣化を防ぐために、輝度の動きベクトルMVの図中円で囲まれた数字で表される値1, 5, 9から1を引いた値を輝度の動きベクトルMVとして用いて、動き補償を行う。その結果、図37の(b)に示しているよう

に、一つの輝度の動きベクトルMVに対して、MPEG 1とMPEG 4のクロマの輝度が一致する。すなわち、入力された画像をMPEG 1画像符号化器120で符号化する際に、検出された輝度のMVが1, 5, 9, ...である場合、それらから1を引いた値を輝度の動きベクトルMVとして用いて、動き補償を行う。もしくは、MPEG 1画像符号化器120では、検出された輝度の動きベクトルMVが1である場合、動きベクトルを0、すなわち動き補償無しモード（no MCモード）になるような処理方法も同様の効果が得られ、さらに画像符号化ビットストリームの発生するビット量を少なくできるという利点を持つ。

【0162】次に、本発明の第5の実施の形態について説明する。

【0163】本発明の第5の実施の形態の構成は、基本的には図1に示した第1の実施の形態の構成と同様であるが、第5の実施の形態の場合、イントラAC係数に対

$$\text{dct_recon}[i] = 2 * \text{QF}[i] * \text{q_scale} * \text{weighting matrix} / 16 \quad \text{MPEG1} \quad (31)$$

$$\text{dct_recon}[i] = (2 * \text{QF}[i] + \text{sign}(\text{QF}[i]) * \text{q_scale} + \text{sign}(\text{QF}[i])) \quad \text{MPEG4} \quad (32)$$

より具体的に説明すると、式(31)及び(32)に示したように、イントラのAC係数の逆量子化方法は、MPEG 1とMPEG 4とで異なる。また、図38は、MPEG 1で量子化係数を $q = 30$ として符号化した場合のDCT係数の再構成例を示している。例えば、DCT係数が0, 1, 2, ...のような値である場合、MPEG 1で量子化係数 $q = 30$ で逆量子化がなされた値は、それぞれ0, 60, 120, ...となる。また、MPEG 4で同じ量子化係数($q = 30$)を用いれば、MPEG 4の逆量子化方式に従って求められる値は、0, 90, 150, 210, ...のように、MPEG 1の逆量子化方式で求められた値より30、すなわち $q (= 30)$ の値が多くなる。そのため、量子化係数 q が増えるにつれて、イントラVOPの劣化が著しくなる。そこで、画質を向上させるには、統計的に、DCT係数の集中している部分を検出し、例えばDCT係数が1近辺に集中しているとすると、MPEG 1で逆量子化された値とMPEG 4で逆量子化された値を一致させるため、式(31)と式(32)を用いて計算を行う。これにより、MPEG 4の量子化係数 q は $60 / 90 * 30$ ($q = 20$)となり、MPEG 4での逆量子化による値は、0, 60, 100, ...となり、MPEG 1と同様あるいは近い値となる。なお、それ以外の値は大きく異なってくるが、極めて発生する確率が少ないため、少々劣化しても気にならない程度である。すなわち、当該第5の実施の形態では、MPEG 1のDCT係数が集中している部分を検出し、その部分におけるMPEG 1とMPEG 4の逆量子化される値を一致させるため、MPEG 4で用いる q の変換を行う。

【0165】なお、量子化係数 q の変換を行わないで、MPEG 1画像符号化器120がウェイトニングマトリ

するMPEG 1の逆量子化方法とMPEG 4の逆量子化方法とが異なる。すなわち、図38に示すように、MPEG 1と同様の量子化係数を用いて、MPEG 4で逆量子化すると、MPEG 1と異なった値になる。そこで、第5の実施の形態では、量子化DCT係数が、ある範囲に集中しているという性質を利用し、画像符号化フォーマット変換装置123において、MPEG 1の量子化係数からMPEG 4の量子化係数を求める。例えば、大部分のDCT係数が0, 1, 2, ...に集中しているとした場合、第5の実施の形態の画像符号化フォーマット変換装置123では、その部分の逆量子化値を一致させるため、式(31)及び(32)に基づいて、例えばMPEG 1の量子化係数 $q = 30$ からMPEG 4の量子化係数 $q = 20$ を求める。或いは、ウェイトニングマトリクスを24に設定しても同様の効果が得られる。

【0164】

クスを新たな値に設定しても同様の効果が得られる。その際、式(31)を用いる代わりに、式(8)を用いて計算すると、ウェイトニングマトリクスの値が24となり、量子化係数 q を30から20に変換した方法と同様の効果が得られる。

【0166】

【発明の効果】以上の説明からも明らかなように、本発明の符号化フォーマット変換装置及び方法においては、第1の符号化方式のパラメータを、そのパラメータの値に応じて第2の符号化方式のパラメータに変換し、当該第2の符号化方式のパラメータを用いて、第1の符号化信号から第2の符号化信号を生成することにより、例えば、MPEG 1画像符号化ビットストリームのような第1の符号化信号をMPEG 4画像符号化ビットストリームのような第2の符号化信号に変換する場合に、信号品質を劣化させずに処理量を低減し、且つ、時間遅延を少なくし、さらに装置構成の大規模化をも防止可能となっている。

【0167】また、本発明の符号化信号生成装置及び方法においては、第1の符号化方式にて使用される第1の量子化方法と第2の符号化方式にて使用される第2の量子化方法とが異なるとき、第1の量子化方法で用いるパラメータに代え、第2の量子化方法で用いるパラメータを第1の符号化方式により符号化された信号に含めて第1の符号化信号を生成することにより、例えば、MPEG 1画像符号化ビットストリームのような第1の符号化信号をMPEG 4画像符号化ビットストリームのような第2の符号化信号に変換する際に、信号品質を劣化させずに処理量を低減し、且つ、時間遅延を少なくし、さらに装置構成の大規模化をも防止可能な、第1の符号化信号を生成できる。

【0168】また、本発明の符号化信号生成装置及び方法においては、第1の符号化信号を生成する際に用いた色差の動きベクトルと上記第2の符号化信号を生成する際に用いた色差の動きベクトルとが異なるとき、第1の符号化信号での色差の動きベクトルと第2の符号化信号での色差の動きベクトルとが等しくなるように、第1の符号化信号の輝度の動きベクトルを第2の符号化信号の輝度の動きベクトルに変換することにより、例えば、MPEG1画像符号化ビットストリームのような第1の符号化信号をMPEG4画像符号化ビットストリームのような第2の符号化信号に変換する際に、信号品質を劣化させずに処理量を低減し、且つ、時間遅延を少なくし、さらに装置構成の大規模化をも防止可能な、第1の符号化信号を生成できる。

【図面の簡単な説明】

【図1】本発明の第1の実施の形態のシステム構成例を示すブロック回路図である。

【図2】画像符号化フォーマット変換装置の詳細な構成を示すブロック回路図である。

【図3】MPEG1パラメータをMPEG4パラメータへ変換する流れ（その1）についての説明に用いる図である。

【図4】MPEG1パラメータをMPEG4パラメータへ変換する流れ（その2）についての説明に用いる図である。

【図5】MPEG1パラメータをMPEG4パラメータへ変換する流れ（その3）についての説明に用いる図である。

【図6】MPEG1パラメータをMPEG4パラメータへ変換する流れ（その4）についての説明に用いる図である。

【図7】MPEG1パラメータをMPEG4パラメータへ変換する流れ（その5）についての説明に用いる図である。

【図8】MPEG1パラメータをMPEG4パラメータへ変換する流れ（その6）についての説明に用いる図である。

【図9】MPEG1パラメータをMPEG4パラメータへ変換する流れ（その7）についての説明に用いる図である。

【図10】レジスタ1, 2に入力するパラメータとその変換の説明に用いる図である。

【図11】レジスタ3に入力するパラメータとその変換の説明に用いる図である。

【図12】レジスタ4に入力したパラメータとその変換の説明に用いる図である。

【図13】レジスタ4に入力したパラメータの関係説明に用いる図である。

【図14】レジスタ7, 8に入力するパラメータとその変換の説明に用いる図である。

【図15】レジスタ9に入力するパラメータとその変換の説明に用いる図である。

【図16】レジスタ10, 11に入力するパラメータとその変換の説明に用いる図である。

【図17】レジスタ12, 13に入力するパラメータとその変換の説明に用いる図である。

【図18】レジスタ14に入力するパラメータとその変換の説明に用いる図である。

【図19】レジスタ15に入力するパラメータとその変換の説明に用いる図である。

【図20】レジスタ16に入力するパラメータとその変換の説明に用いる図である。

【図21】レジスタ17, 18に入力するパラメータとその変換の説明に用いる図である。

【図22】マクロブロックのdquantのパラメータの求め方の説明に用いる図である。

【図23】連続した2つのマクロブロックのquantizer_scaleの差分をMPEG4のシンタックスの制約に従って3つのケースに分け、dquantを求める処理の説明に用いる図である。

【図24】ピクチャの先頭のスライスヘッダを除去する処理の説明に用いる図である。

【図25】ピクチャの先頭以外のスライス層の処理の説明に用いる図である。

【図26】レジスタ21に入力するパラメータとその変換の説明に用いる図である。

【図27】レジスタ27に入力するパラメータとその変換の説明に用いる図である。

【図28】レジスタ23, 24, 25, 26に入力するパラメータとその変換の説明に用いる図である。

【図29】MPEG1とMPEG4の動きベクトルの予測方法の説明に用いる図である。

【図30】レジスタ28, 29, 30, 31に入力するパラメータとその変換の説明に用いる図である。

【図31】MPEG1とMPEG4のイントラマクロブロックのDC係数の予測方法の説明に用いる図である。

【図32】MPEG1の2次元のラン、レベルペアと、MPEG4の3次元のラン(run)、レベル(level)、ラスト(last)ペアの関係を示す図である。

【図33】リシンクマーカを配置した後の動きベクトルの変換の説明に用いる図である。

【図34】イントラマクロブロックのDC係数の変換の説明に用いる図である。

【図35】MPEG4のDC係数の量子化係数とAC係数の量子化係数の関係説明に用いる図である。

【図36】式(26)～式(29)を論理回路で演算する場合の近似式の説明に用いる図である。

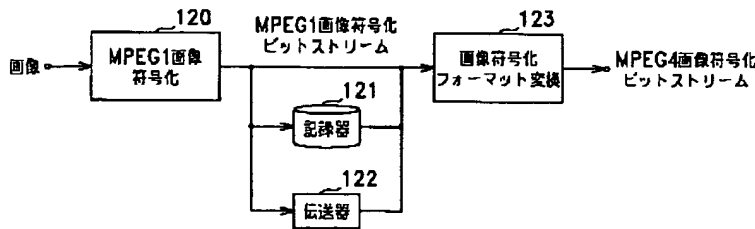
【図37】本発明の第4の実施の形態における輝度の動きベクトル処理の説明に用いる図である。

【図38】本発明の第5の実施の形態の場合に、イント

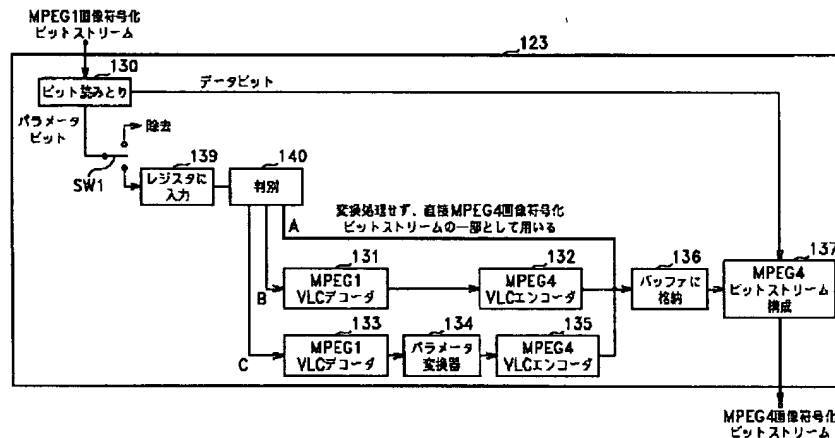
ラAC係数に対するMPEG1の逆量子化方法とMPEG4の逆量子化方法の説明に用いる図である。

【図39】従来のシステム構成例を示すブロック回路図である。

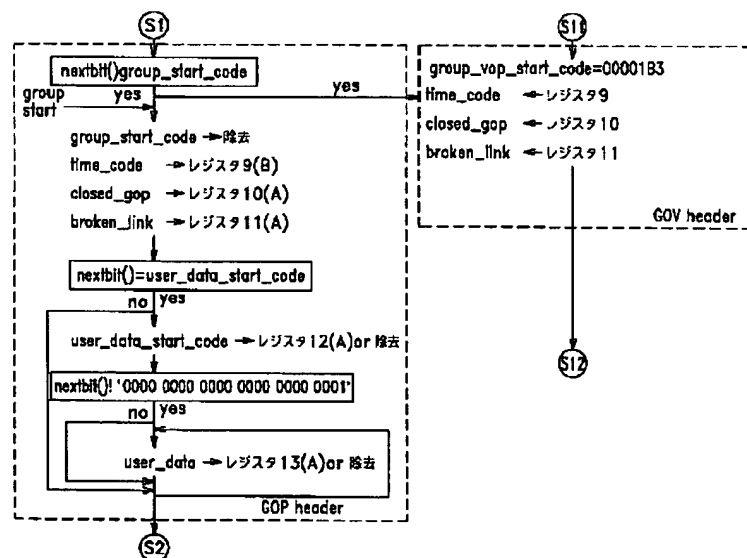
【図1】



【図2】



【図4】



【符号の説明】

120 MPEG1画像符号化装置、121 記録器、122 伝送器、123 画像符号化フォーマット変換装置

【図15】

MPEG1 (time_code)	MPEG4 (time_code)
drop_frame_flag	除去
time_code_hours	time_code_hours
time_code_minutes	time_code_minutes
marker_bit	marker_bit
time_code_seconds	time_code_seconds
time_code_pictures	除去

【図19】

full_pel_forward_vector	MPEG4 の動きベクトル
0	MPEG1 の動きベクトルと同様
1	MPEG1 の動きベクトルの2倍

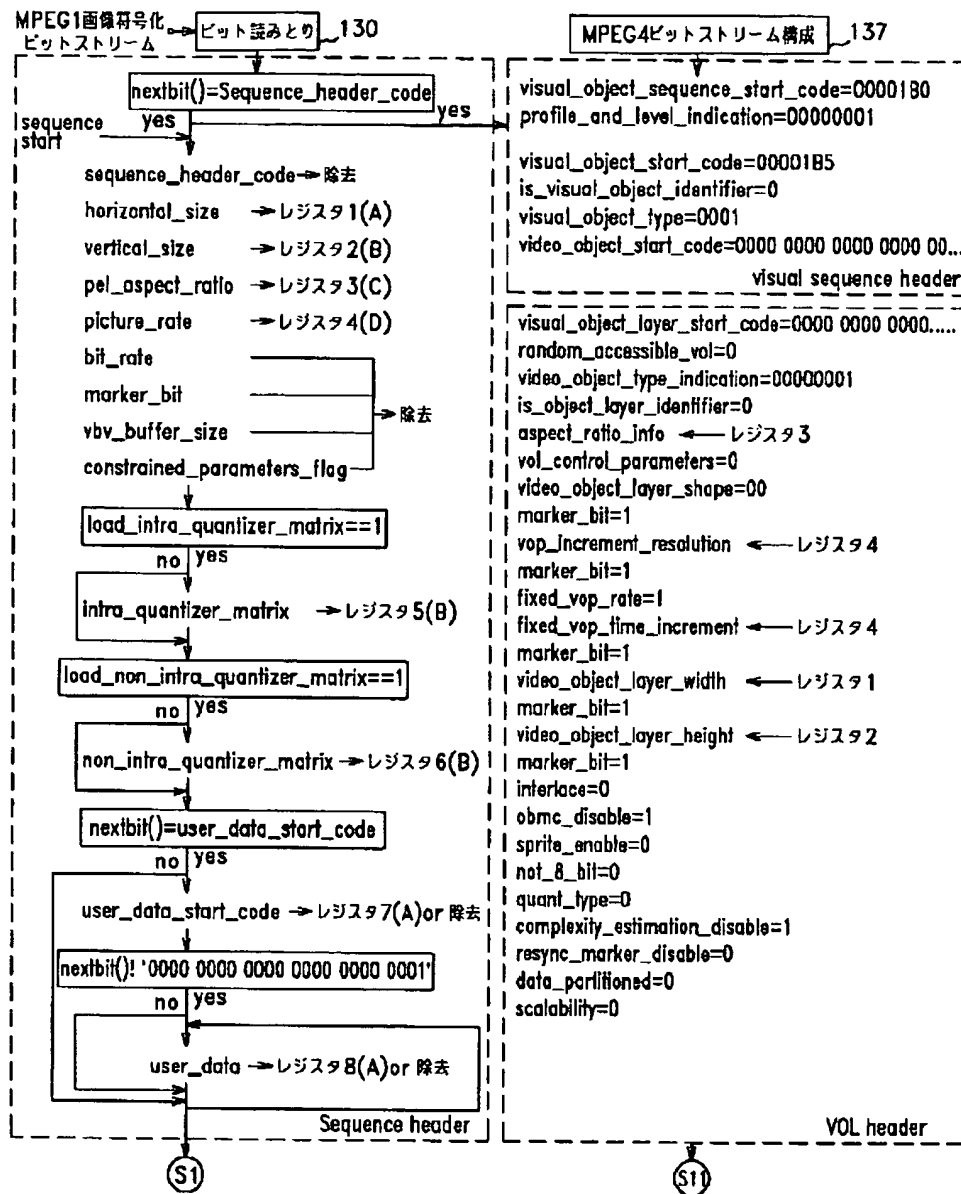
【図10】

	MPEG1	MPEG4
レジスタ1	horizontal_size	video_object_layer_width
レジスタ2	vertical_size	video_object_layer_height

【図11】

MPEG1 (pel_aspect_ratio)	MPEG1 VLCデコーダ	MPEG4VLCエンコーダ (aspect_ratio_info)
0001	1:1	0001
0011	16:9 625line	0100
0110	16:9 525line	0101
1000	4:3 625line	0010
1100	4:3 525line	0011
その他		extended PAR

【図3】



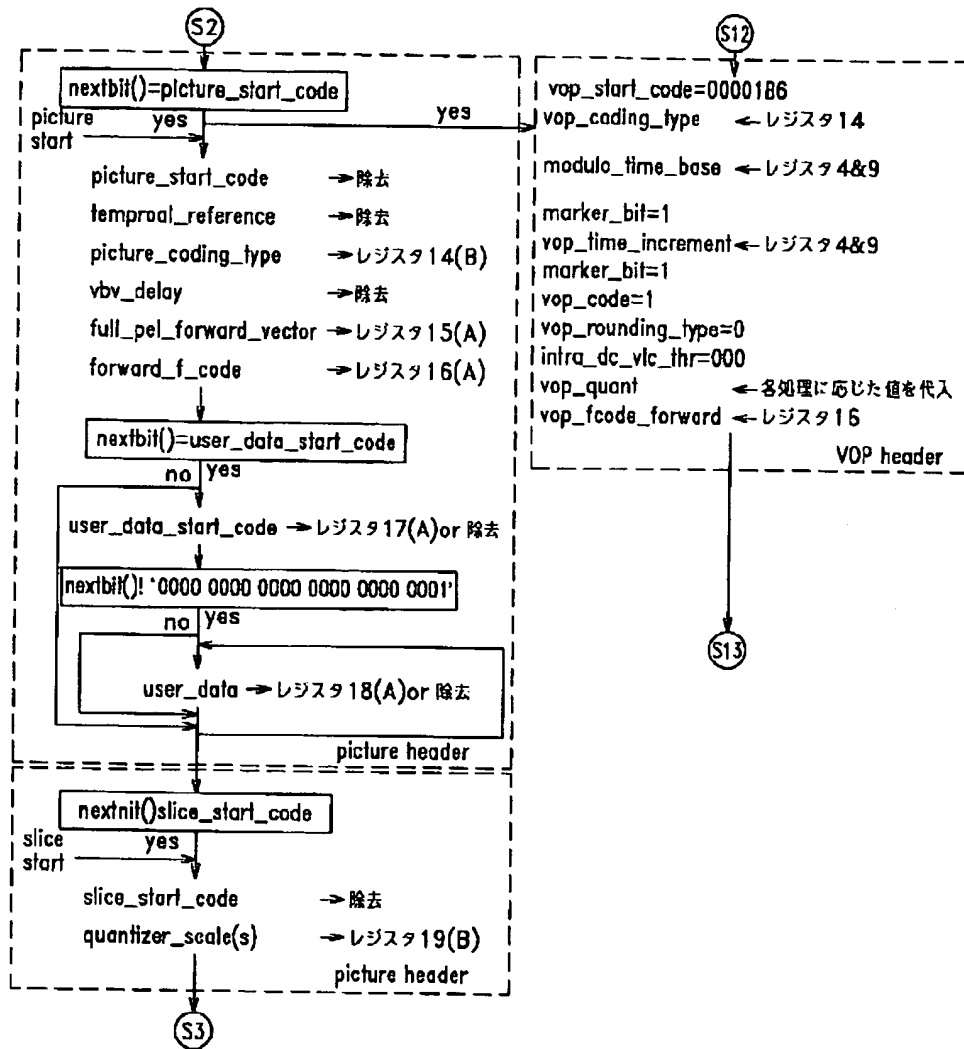
【図12】

MPEG1 frame_rate_code	MPEG1 VLCコード	MPEG4 vop_time_increment_resolution	MPEG4 vop_time_increment
0001	23.976	24000	0,1001,2002,3003...n
0010	24	24	0,1,2,3,4,5...n
0011	25	25	0,1,2,3,4,5...n
0100	29.97	30000	0,1001,2002,3003...n
0101	30	30	0,1,2,3,4,5...n
0110	50	50	0,1,2,3,4,5...n
0111	59.94	60000	0,1001,2002,3003...n
1000	60	60	0,1,2,3,4,5...n

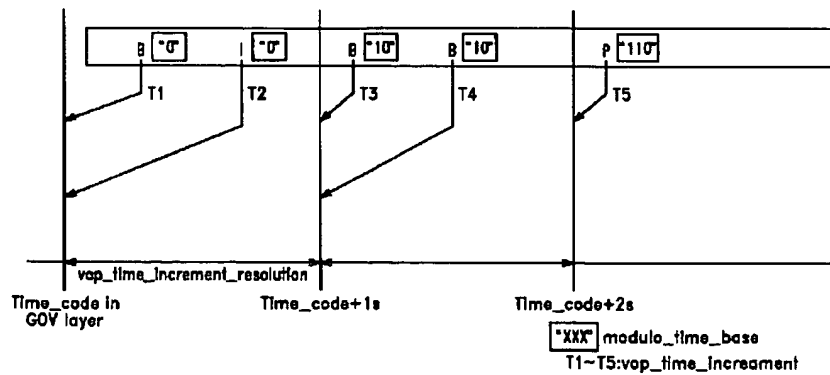
【図27】

MPEG1	MPEG4
pattern_code[0~3]	cbpy
pattern_code[4,5]+mbtype	mcbpc

【図5】



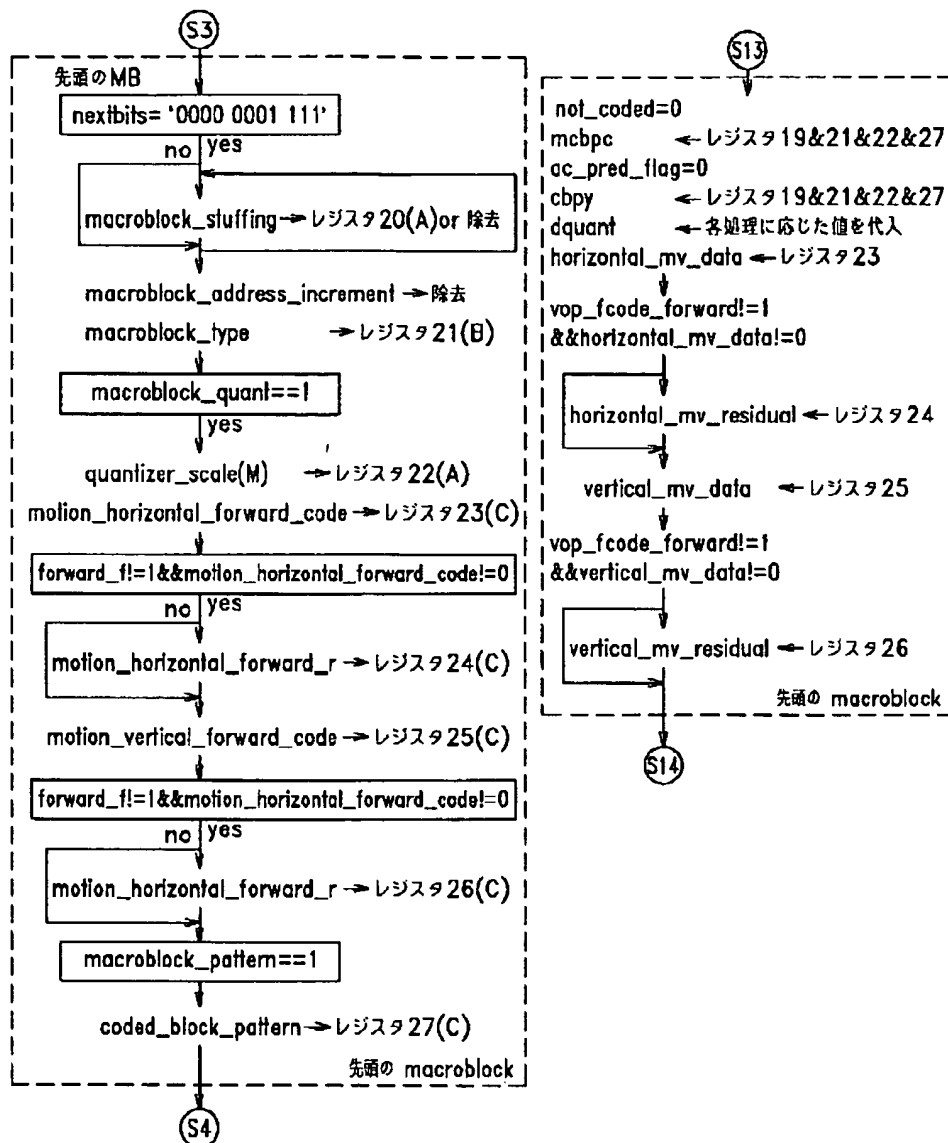
【図13】



【図20】

	MPEG1	MPEG4
レジスタ16	forward_f_code	vop_fcode_forward

【図6】



【图 14】

	MPEG1	MPEG4
レジスタ7	user_data_start_code	user_data_start_code
レジスタ8	user_data	user_data

【图 17】

	MPEG1	MPEG4
レジスタ12	user_data_start_code	user_data_start_code
レジスタ13	user_data	user_data

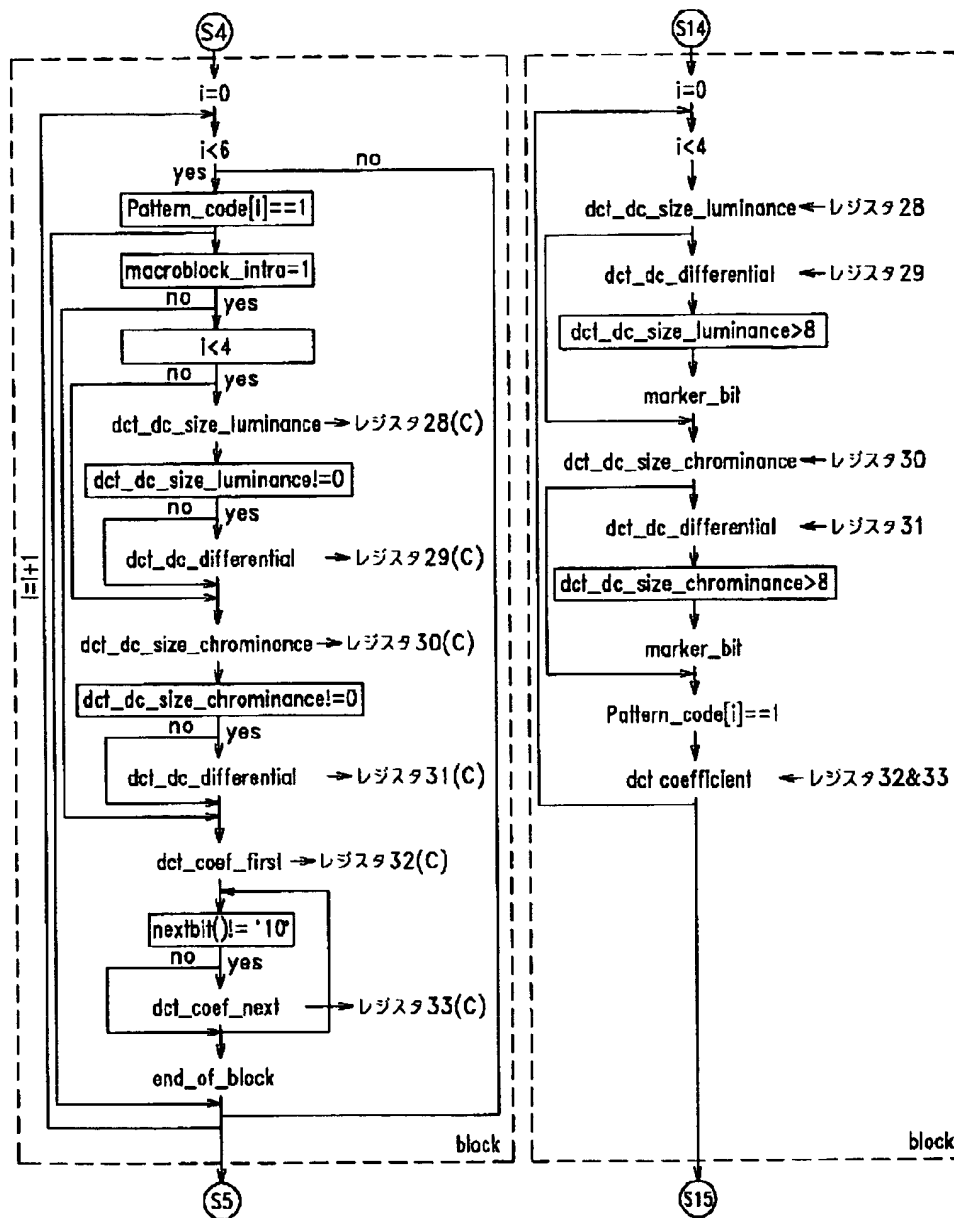
【图 16】

	MPEG1	MPEG4
レジスタ10	closed_gop	closed_gov
レジスタ11	broken_link	broken_link

【図 18】

MPEG1 (picture_coding_type)	MPEG1 VLCデコーダ	MPEG4VLCエンコーダ (vop_coding_type)
001	I	00
010	P	01
011	B	10

【図 7】



【図 2 1】

【図 2 3】

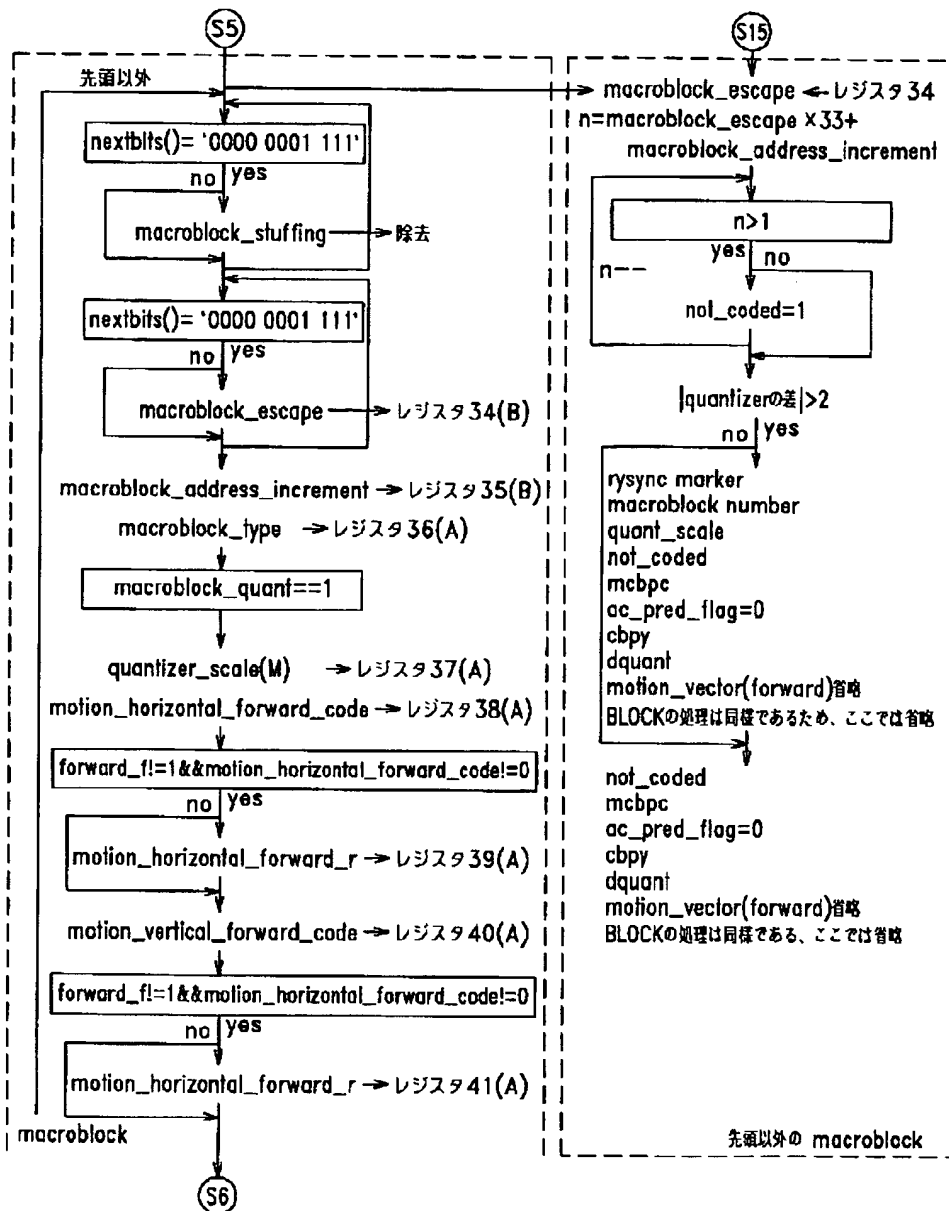
	MPEG1	MPEG4
レジスタ17	user_data_start_code	user_data_start_code
レジスタ18	user_data	user_data

case 1	case 2	case 3
差分=1,-1,2,-2	差分=0 すなわち quantizer_scale が送られていない場合	差分の絶対値>2
dquant=差分	mbtype による dquant のフラグを依にする	対象となる 4日 の前に、resync marker を配置する

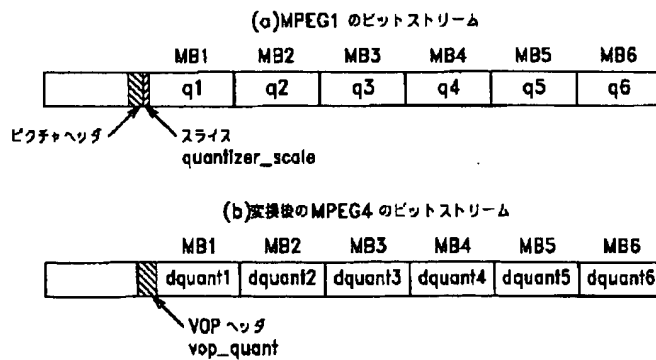
【図 3 2】

MPEG1	MPEG4
run[i],level[i]	last=0,run[i],level[i]
最後のrun[i],level[i]	last=1,run[i],level[i]

【図8】



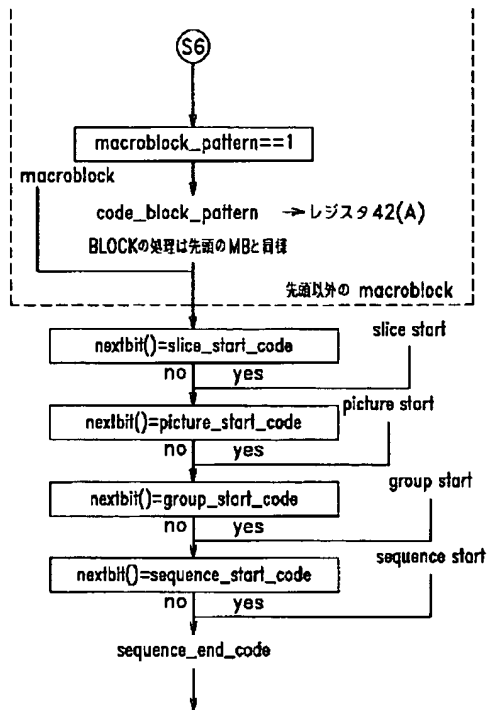
【図24】



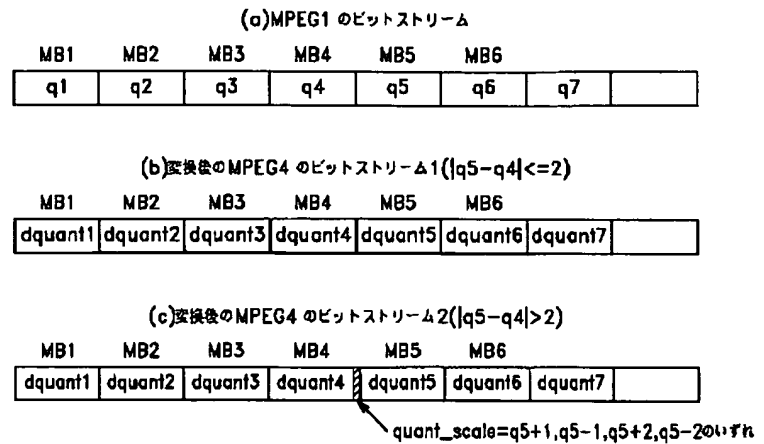
【図26】

MPEG1 (macroblock_type)	MPEG1 VLCコード	MPEG4 (mbtype)
1	Intra I	3
01	Intra+q I	4
1	Inter P	0
01	Stuffing	Stuffing or 除去
001	Stuffing	Stuffing or 除去
00011	Intra P	3
00010	Inter+q P	1
00001	Stuffing	Stuffing or 除去
000001	Intra+q P	4

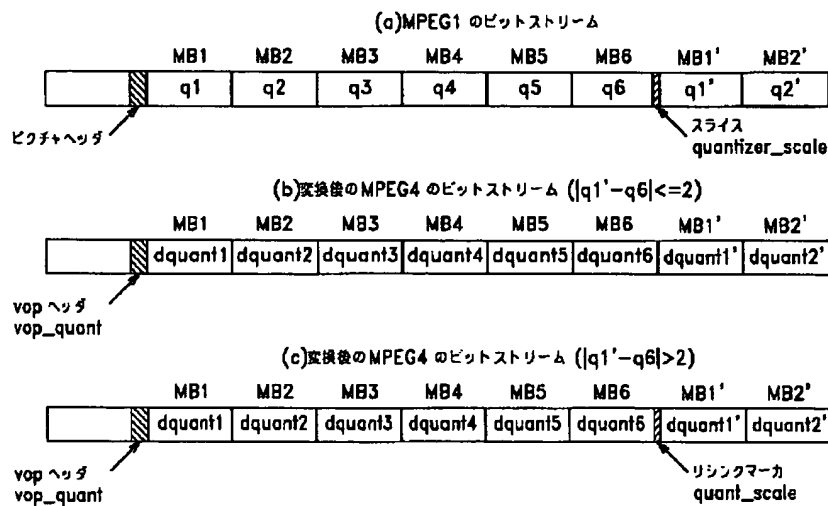
【図 9】



【図 2 2】



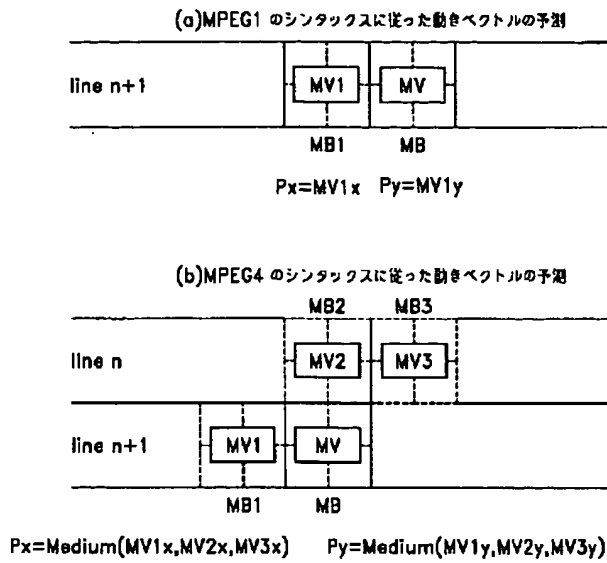
【図 2 5】



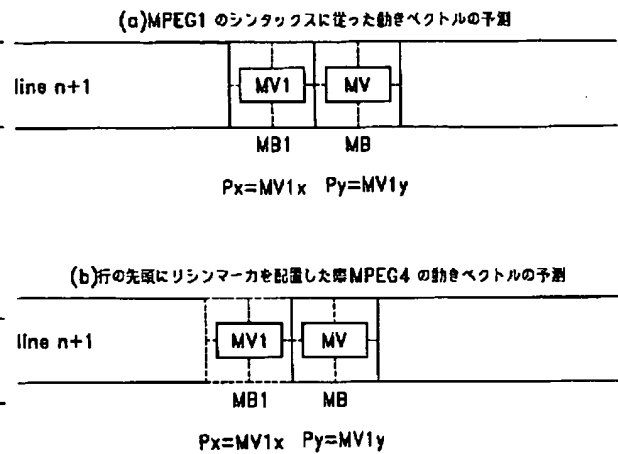
【図 2 8】

	MPEG1	MPEG4
レジスタ 23	motion_horizontal_forward_code	horizontal_mv_data
レジスタ 24	motion_horizontal_forward_r	horizontal_mv_residual
レジスタ 25	motion_vertical_forward_code	vertical_mv_data
レジスタ 28	motion_vertical_forward_r	vertical_mv_residual

【図 29】



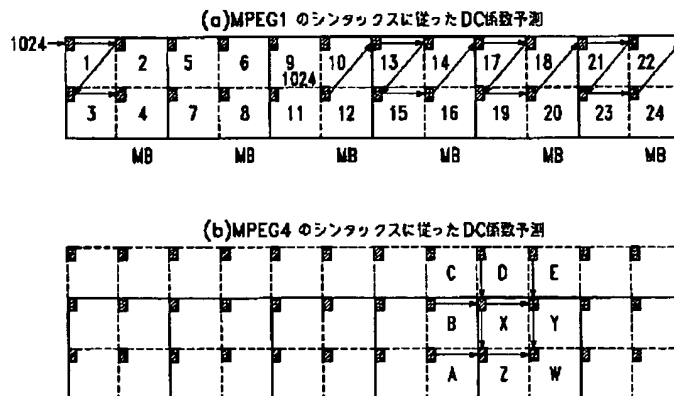
【図 33】



【図 30】

	MPEG1	MPEG4
レジスタ 28	dct_dc_size_luminance	dct_dc_size_luminance
レジスタ 29	dct_dc_differential	dct_dc_differential
レジスタ 30	dct_dc_size_chrominance	dct_dc_size_chrominance
レジスタ 31	dct_dc_differential	dct_dc_differential

【図 31】



【図 38】

例 DCT係数を(q30,w16)で符号化した場合の再構成例

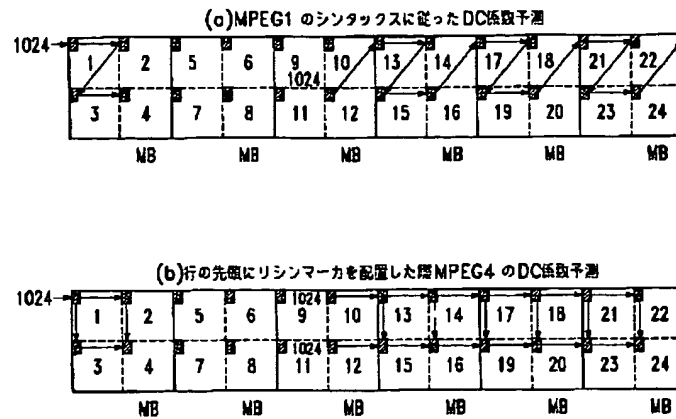
量子化DCT係数	0	1	2	3	4	5
MPEG1での逆量子化(q=30)	0	60	120	180	240	300
MPEG4での逆量子化(q=30)	0	90	150	210	270	330
MPEG4での逆量子化(q=20)	0	60	100	140	180	220

大部分のDCT係数はこの部分に集中

【図 35】

Component: Type	Dc_scaler for quantizer_scale range			
	1 through 4	5 through 8	9 through 24	>=25
Luminance: Type1	8	$2 \times \text{quantizer_scale}$	$\text{quantizer} + 8$	$2 \times \text{quantizer_scale} - 16$
Chrominance: Type2	8	$(\text{quantizer_scale} + 13) / 2$	$\text{Quantizer_scale} - 6$	

【図34】



【図36】

AC 係数の q	輝度の DC係数の q	8/q	近似式	クロマの DC係数の q	8/q	近似式
5	2×5	$\frac{4}{5}$ =0.8	$\frac{51}{64}$ = $\frac{(32+16+2+1)}{64}$ =0.797	18/2	$\frac{8}{9}$ =0.889	$\frac{57}{64}$ = $\frac{(32+8+4+2+1)}{64}$ =0.891
6	2×6	$\frac{2}{3}$ =0.667	$\frac{43}{64}$ = $\frac{(32+8+2+1)}{64}$ =0.672	19/2	$\frac{16}{19}$ =0.842	$\frac{27}{32}$ = $\frac{(16+8+2+1)}{32}$ =0.844
7	2×7	$\frac{4}{7}$ =0.571	$\frac{37}{64}$ = $\frac{(32+4+1)}{64}$ =0.578	20/2	$\frac{4}{5}$ =0.8	$\frac{51}{64}$ = $\frac{(32+16+2+1)}{64}$ =0.797
8	2×8	1/4	1/4	21/2	$\frac{16}{21}$ =0.762	$\frac{49}{64}$ = $\frac{(32+16+1)}{64}$ =0.766

【図37】

(a) クロマの動きベクトル

輝度の MV	0	①	2	3	4	⑤	6	7	8	⑨	...
MPEG1(輝度の MV/2)	0	①	1	1	2	②	3	3	4	④	...
MPEG4(MPEG4 シンタックス)	0	①	1	1	2	③	3	3	4	⑤	...

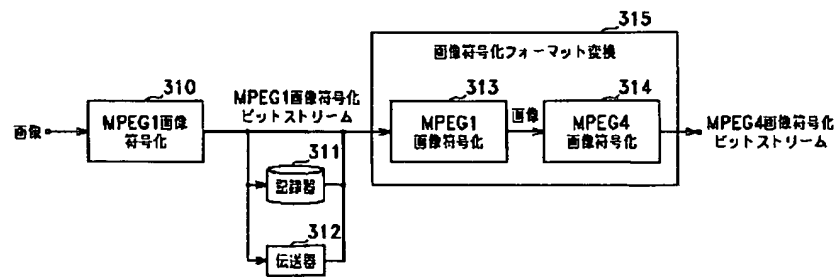
検出された輝度の MV=1,5,9,13,....



(b) 符号化時に用いる輝度の MV' = 検出された輝度の MV-1

輝度の MV'	0	①	2	3	4	④	6	7	8	⑧	...
MPEG1(輝度の MV/2)	0	①	1	1	2	②	3	3	4	④	...
MPEG4(MPEG4 シンタックス)	0	①	1	1	2	②	3	3	4	④	...

【図 3 9】



フロントページの続き

(72)発明者 名雲 武文
東京都品川区北品川 6 丁目 7 番 35 号 ソニ
ー株式会社内

F ターム(参考) 5C059 KK01 KK41 NN01 PP04 RC11
RC16 RC24 SS01 SS07 SS11
SS12 SS13 UA02 UA05 UA32
UA36
5J064 BA09 BA16 BC02 BC04 BC16
BD02 BD03